공학박사학위논문

협소하고 복잡한 환경에서 자율 주행을 위한 샘플링 기반 모션 계획 방법

Sampling-based Motion Planning Approaches for Autonomous Vehicle in Narrow Cluttered Spaces

2017년 8월

서울대학교 융합과학기술대학원
융합과학부 지능형융합시스템전공
신세호

협소하고 복잡한 환경에서 자율 주행을 위한 샘플링 기반 모션 계획 방법

Sampling-based Motion Planning Approaches for Autonomous Vehicle in Narrow Cluttered Spaces

지도교수 박재흥

이 논문을 공학박사 학위논문으로 제출함

2017년 8월

서울대학교 융합과학기술대학원

융합과학부 지능형융합시스템전공

신세호

신세호의 박사 학위논문을 인준함

2017년 8월

위 원 장 곽노준 (인)

부위원장 박재흥 (인)

위 원 문형필 (인)

위 원 안정호 (인)

위 원 정우진 (인)

Abstract

Sampling-based Motion Planning Approaches for Autonomous Vehicle in Narrow Cluttered Spaces

Seho Shin
Department of Transdisciplinary Studies
The Graduate School of Convergence Science and Technology
Seoul National University

Autonomous vehicles are being actively developed for fully autonomous driving without driver intervention. Motion planning is one of the most key technologies in terms of driving safety and efficiency. In particular, the motion planning in constrained narrow space such as a parking lot is very challenging because it requires many changes in forward and backward directions and adjustments of position and orientation of the vehicle. In this thesis, a sampling-based motion planning algorithm is proposed based on Rapidly-exploring Random Trees (RRT, RRT*) by specifying desired orientation during the tree expansion and the rewiring step. The contribution is as follows. First, an efficient sampling method is proposed for narrow-cluttered area. In this area, the probability of obtaining a sample to pass

through the area due to the obstacle area is relatively low than an open area. It may also fail to extend the path if sampled position is difficult to extend from adjacent nodes. To solve this problem, a constraint model on the tangential direction of the random sample is proposed. Second, we propose an extension method based on tangential direction constraint. In the process of expanding the tree to random samples, a large number of nodes in narrow-cluttered regions cannot pass the collision test. This increases unnecessary iteration numbers and increases memory usage. To solve this problem, we propose a node extension method based on gradient descent. The proposed algorithm has been tested in various situations and its results demonstrated much faster target path search and convergence to the optimal path than the existing nonholonomic RRT*.

Keywords : Autonomous Vehicle, Motion Planning, Rapidly-exploring Random Tree, Nonholonomic Path Planning

Student Number : 2010-22689

Contents

List of Figures

List of Tables

List of Algorithms

Chapter. 1

Introduction

1.1 Autonomous Vehicles

An autonomous vehicle means that the vehicle itself recognizes the driving environment and operates the control device without human intervention to its destination. With the introduction of autonomous vehicles, deaths annually in traffic accidents (US 33,000, China 260,000) is expected to decrease significantly [1]. Productive works are also possible by reducing the burden of driving. Based on these expectations, development is underway with the aim of commercialization in 2020. Since 2035, vehicles equipped with autonomous driving systems are expected to account for about 75% of total vehicle sales [2].

NHTSA(National Highway Traffic Safety Administration) has developed a five-level classification of autonomous vehicles and provides specific performance-based definitions for each of the five levels (Figure.1.1). Level 0 is not automated. Level 1 is an automation phase with Lane Keeping Assist System (LKAS) or Adaptive Cruise Control (ACC). Level 2 is a combination of two or more control functions, such as Highway Assist (HAD) or Traffic Disaster Support (TJA). Level 3 refers to the autonomous driving in restricted environments such as a highway, where continuous operation is not required. Level 4 can be completely autonomous in all situations.

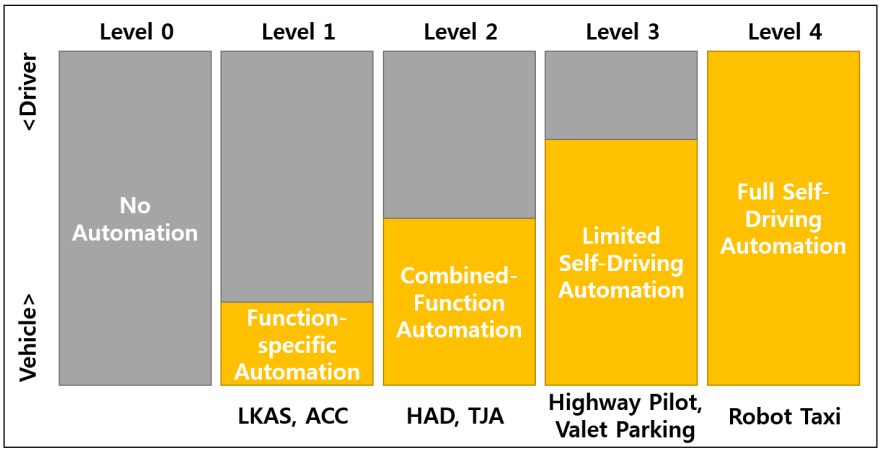


Figure. 1.1: Five levels of driving automation (NHTSA)

Companies developing autonomous driving technology are currently testing technology levels above level 3. Mercedes-Benz, Tesla, Volvo, and Hyundai are applying Level 2 technology to the market. Google has tested over 3 million kilometers of autonomous vehicles at level 4.

1.2 Planning System of Autonomous Vehicles

Techniques for autonomous driving can be divided into four categories: perception, ego-vehicle localization, planner, and controller. The perception is a technology that analyzes sensor data and extracts useful information and contexts in the driving environment. Ego-vehicle localization estimates the position of a global or local vehicle based on a GPS or distance measurement sensor. The planner is the process of determining behavior to meet target behavior using perception results and ego-vehicle localization information. The controller generates a steering and velocity inputs to follow the planned

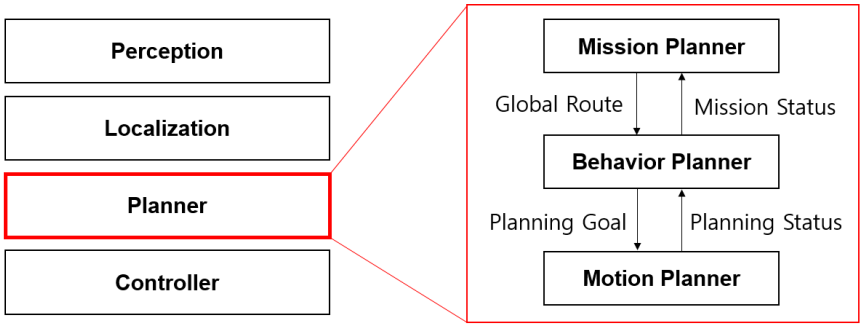


Figure. 1.2: System architecture of an autonomous vehicle

path.

At level 2, the vehicle-centered local path generation plays an important role for ACC and LKAS. However, as the level of the autonomous driving system increases to 3 or 4, a more intelligent vehicle planner system is required. At the level 3 or above, it is required to generate a route to arrive at the destination in consideration of the global information of lanes, the intersection, the stop sign, and the parking area, etc. The behavioral decision making is also necessary to cope with various situations that can occur on the road, such as lane change, overtaking, and emergency stop. In addition, a robust motion planner is required to generate a safe path to uncertainties around the vehicle when performing determined behaviors. The path planning system can be classified into the following three types (as shown in Figure.1.2) : Mission Planner, Behavior Planner, Motion Planner. The mission planner generates a global route using a pre-defined road information database such as a RNDF (Route Network Definition File). The behavior planner determines the planning goal. The motion planner computes the lo-

Figure. 1.3: Narrow and complex parking lot: the complexity of the path planning is highly due to the wide configuration space while the narrow drivable area.

cal path to carry out the given goal and transmits it to the control system. This thesis is concerned with motion planner problems to improve driving safety and efficiency even in high uncertainty situations such as general urban environments.

1.3 Contribution of Thesis

The contribution of this thesis is as follows; first, an efficient sampling method is proposed for a narrow-cluttered area in which the probability of obtaining a sample that can pass through the area despite the various obstacles it contains is relatively low compared to an open area. It may also fail to extend the path if the sampled position is difficult to extend from adjacent nodes. A constraint model on the tangential direction of the random

sample is proposed to solve this problem. Second, we propose an extension method that is based on the tangential direction constraint. In the process of expanding the tree to random samples, a large number of the nodes in narrow cluttered regions cannot pass the collision test. This increases both the unnecessary iteration number and memory usage. We have tried to solve this problem with a proposed node extension method that is based on gradient descent.

This thesis is organized as follows. Chapter 2 is a literature review of motion planners. In Chapter 3, a sampling-based kinodynamic motion planning algorithm is proposed for narrow cluttered environments, Chapter 4 contains a proposed sampling-based optimal motion planning algorithm for narrow cluttered environments, Chapter 5 introduces the development and validation of an experimental platform for autonomous valet parking, and Chapter 6 concludes the study.

Chapter. 2

Related Works

2.1 Motion Planning for Aunomous Vehicles

The goal of autonomous vehicles is fully autonomous driving in our daily life. A great deal of study in various research fields have been conducted for the fully automated driving without driver intervention. To achieve fully autonomous driving not only on a highway but also in various complicated and unpredictable situations in a city, the safe and efficient path planning is one of the major issues to be considered. Path planning methods for autonomous vehicles can be divided into the following two categories: the method for a structured environment, and the method for an unstructured(or semi-structured) environment [3, 4]. The structured environment is a place with lane information such as highways or public roads. The motion planning method for the structured environment deals mainly with behavior decisions such as lane keeping, lane change, and intersection scenarios. On the other hand, the unstructured (or semi-structured) environment is an open or rugged area with no lane information, such as a parking lot. In unstructured environments, motion planning addresses the problem of finding a collision-free path based on drivability consideration. This problem has relatively high computational complexity because of the various uncertainties that occur in a large search space [5]. The motion planning approach for

unstructured environments can be divided into a grid-based method, a potential field-based method, and a sampling-based method according to the workspace representation.

The grid-based method decomposes the workspace into a set of subsets as a grid and stores the discretization state for each grid. A* and D* are representative algorithms in this category [6, 7]. Hybrid A* and D* has been also proposed for vehicle path generation considering constraints of the nonholonomic system [8, 9]. This approach has been used to identify the optimal path within the grid space. However, a discontinuous path can be generated in discretizing the state of the robot using the motion primitive. For this reason, an additional computation is required to search the path in a high-dimensional configuration. Especially, in narrow and cluttered environments, high resolution grids and high quality motion primitives are required for the precise path generation. This can exponentially increase the computational time. In the potential field-based approach, a collision-free path is identified by moving from high- to low-potential energy [10, 11]. This approach can thus quickly identify a collision-free path. A potential field model for the lane has been proposed and applied to an autonomous driving system. However, this approach is difficult to model the kinematics and dynamics characteristics of the system. It is also easy to fall into a local minima in a narrow and cluttered area.

The sampling-based method generates the state information of the system through sampling and expands the node considering the connectivity of the configuration space. The most commonly used algorithms are Probabilistic Roadmap Method (PRM) and Rapidly-exploring Random Trees

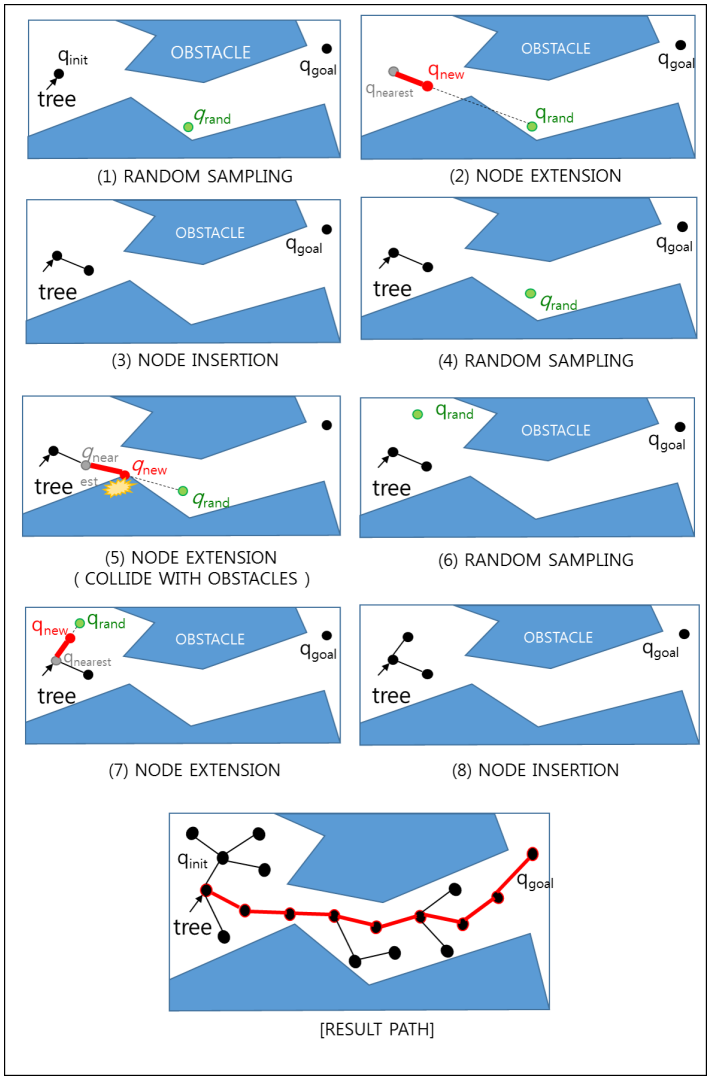


Figure. 2.1: Procedure of the RRT algorithm

(RRT) [12].The PRM consists of a learning phase and a query phase [13, 14]. In learning phase, a probabilistic roadmap is constructed offline and then paths are extracted online in a query phase. This method has a disad-

vantage that it is difficult to create the roadmap in dynamic environments. The RRT method searches the path by expanding the tree considering the connectivity of the randomly generated nodes. Figure.2.1 represents the procedure of the RRT algorithm. This method has the disadvantage that the quality of the path can vary depending on the random sample. Nevertheless, it is widely employed because it is useful for high-dimensional configuration space and easy to reflect constraint conditions of systems.

2.2 Sampling-based Motion Planning Algorithms

The basic version of the RRT algorithm has been extended to various academic fields; RRT-CONNECT connects trees between starting and target points through a simple greedy heuristic [15], RRT-BLOSSOM adopts a flood-fill-like mechanism into multiple expansions [16], and T-RRT improves path quality by following valleys or saddle points and applies a cost map to expand the tree [17]. Path quality can be improved using the RRT* to find the optimal path through tree refinements [18]. With respect to car-like models, kinodynamic RRT, kinodynamic RRT*, and TP-RRT methods reflect nonholonomic constraints [19, 20, 21], while CL-RRT and fast RRT methods consider road characteristics for autonomous driving vehicles [22, 23]. Methods of calculating the optimal path by applying the dynamics model of the vehicle system have also been proposed [24, 25, 26]. However, these methods are less useful in the event that a space is very narrow, or when a vehicle's accurate position and orientation are required near obstacles because the probability is very low that random nodes are located in a

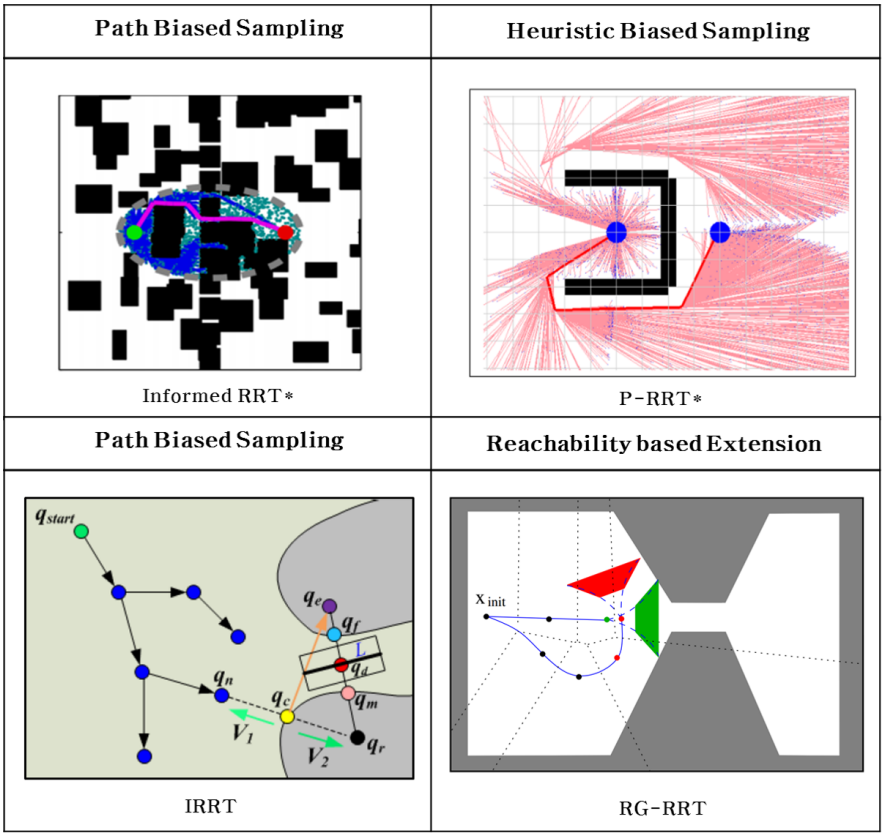


Figure. 2.2: Review of sampling-based algorithms for narrow and cluttered environments

narrow region with the vehicle's desired position [27, 28]. A sampling-based approach and an expansion-based approach have been proposed to improve the performance of RRT-based planners in a narrow space. The sampling-based approach can be divided into a biased sampling strategy and a retraction strategy for sampling where a large number of nodes are distributed in a narrow region. Unlike uniform sampling methods, the biased sampling strategy allows for high-density sampling by assigning a high probability

to regions of interest. This strategy can be categorized as either heuristic-biased sampling or path-biased sampling according to the information used. The heuristic-biased strategy performs sampling in a low-cost direction by taking heuristic estimates of nodes into account [29, 30, 31, 32, 33]. This method has the advantage of shortening the search time by extending the tree to the target position much more rapidly than the uniform sampling method. However, the inaccuracy of the empirical estimates in the local minimum area can cause performance degradation. The optimal path can not be guaranteed when violating a uniform sampling assumption to ensure the asymptotic optimality of RRT*. Path-biased sampling is a method of increasing the frequency of random samples around a feasible solution [34, 35, 36]. It is typically applied to the RRT algorithm by considering anytime characteristics, so the path's quality is improved as the iterations increase. This can also be applied to RRT* to perform sampling within a limited range of the initial generation path to search for the optimal path efficiently. The path-biased sampling method has the advantage of minimizing the performance degradation caused by expanding of the search range. However, the feasible path must be searched before it can be applicable. The retraction biasing strategy creates samples uniformly and then moves them to valid areas; one of these methods moves the samples taken in an obstacle area to a narrow area using a bridge-line test [37, 38, 39, 40]. Preprocessing is required to retract samples to a feasible region, but this is useful for applying optimal path planning methods by satisfying a uniform sampling assumption for asymptotic optimality. The node extension-based approach has been proposed to reduce the amount of computation time and memory overhead caused by process-

ing invalid nodes in narrow environments. Selectively expansion methods have been proposed, e.g., a method of deactivating non-extensible nodes, a method of determining an extension direction that considers obstacles, and a method of applying exploration and exploiting the ratios of nodes [41, 42]. There is also a method of expanding a node based on the reachability analysis for each [43, 44]. Figure.2.3 illustrates the proposed RRT and RRT* methods for narrow and cluttered environments. However, research to improve performance in narrow spaces is still lacking for car-like robots. This study proposes a method of conducting effective path planning in a narrow space. In particular, the parking path in a narrow target space is a challenging issue that requires several forward and backward vehicle movements.

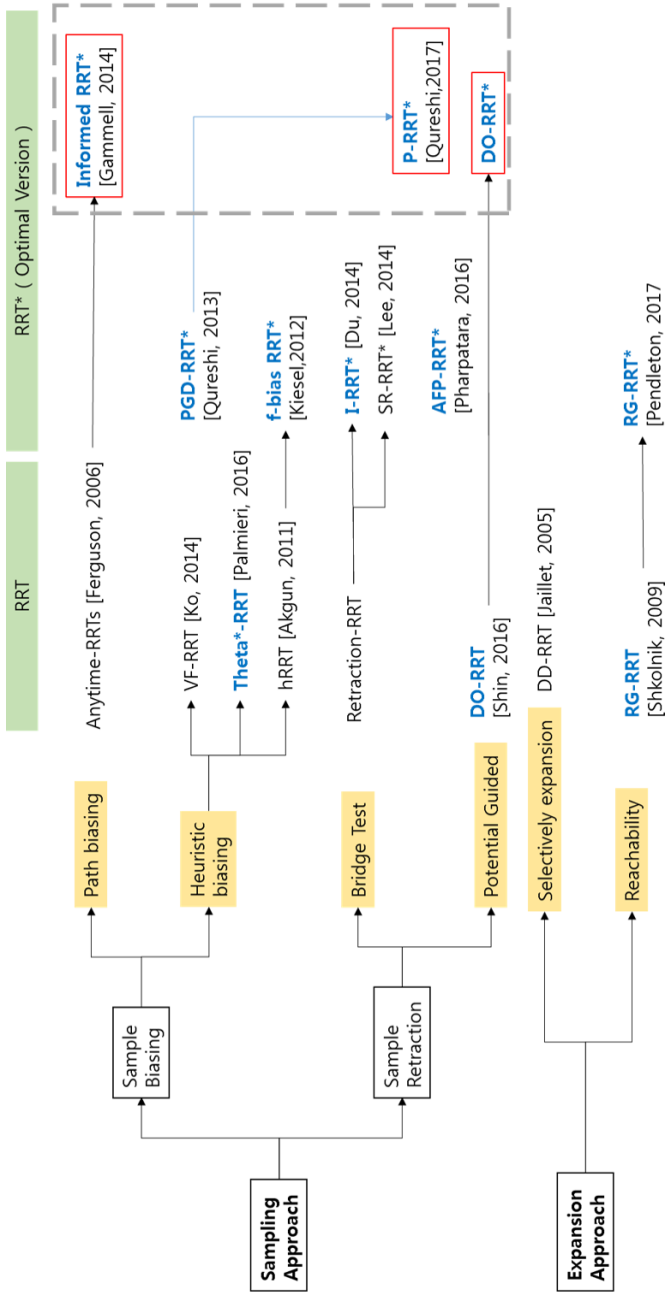


Figure. 2.3: Sampling based RRT/RRT* algorithms for narrow and cluttered environments

Chapter. 3

Sampling-based Kinodynamic Motion Planning Algorithm for Narrow Cluttered Environments

3.1 Overview

In this chapter, a sampling-based kinodynamic motion planning algorithm is proposed for narrow and complex environments. In particular, planning a path in a narrow target space, such as a parking lot, is a challenging issue that requires several forward and backward vehicle movements. To overcome this limitation, the desired orientation RRT (DO-RRT) is proposed. This algorithm models nonholonomic constraints of the vehicle and geometric constraints of obstacles on the tangential vector space and extends a tree using this tangential vector as a desired orientation of random samples. This method is verified through the comparison with the nonholonomic RRT algorithm in parking scenarios.

3.2 Preliminary Definition

3.2.1 Problem Statements

The configuration and control spaces for the path-plan problems are defined as $Q \subset R^m$ and $U \subset R^n$ respectively. A nonholonomic vehicle system can be expressed with the following differential equation.

$$\dot{q}(t) = f(q(t), u(t)), q(0) = q_{init} \tag{3.1}$$

where $q(t) \in Q$ and $u(t) \in U$ are states and inputs of the vehicle respectively; $q_{init} \in Q$ is the initial condition, where m and n are the dimension of the initial condition's states and inputs, respectively; and f is the function of vehicle's kinematics or dynamics. In the configuration space, $Q_{obs} \subset Q$ is the space with obstacles and $Q_{free} \subset Q \setminus Q_{obs}$ is defined as a free space without obstacles. A planning problem can be formulated as follows.

$$\begin{aligned} q(t) \in Q_{free}, \forall t \in [0, t_f] \\ \dot{q}(t) = f(q(t), u(t)), \\ q(0) = q_{init}, \qquad (3.2) \\ q(t_f) = q_{goal}, \qquad (3.3) \\ u(t) \in U, \forall t \in [0, t_f] \end{aligned}$$

where q_{goal} is a target position and orientation, and t_f is the time to reach the goal (q_{goal}). The solution path consists of a control input u, a time interval t, and a state that can be obtained sequentially by the integration of

the kinodynamic function f. Due to computational complexity issues, many methods is used a approximated forms or database of candidate paths.

3.2.2 Autonomous Vehicle Model

In this study, rear-wheel driving model is considered as a vehicle's kinematics [45]. The model is expressed as follows.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \\ \tan(\delta)/L \\ 0 \end{bmatrix} \overrightarrow{u}_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \overrightarrow{u}_2, \tag{3.4}$$

where x and y indicate vehicle's position, θ is a vehicle's heading angle, L represents the distance between the front and rear axles of the vehicle, and δ denotes the steering angle. Control inputs u_1 and u_2 are the longitudinal and angular velocities of the steering wheel, respectively.

3.3 Kinodynamic RRT and Limitations

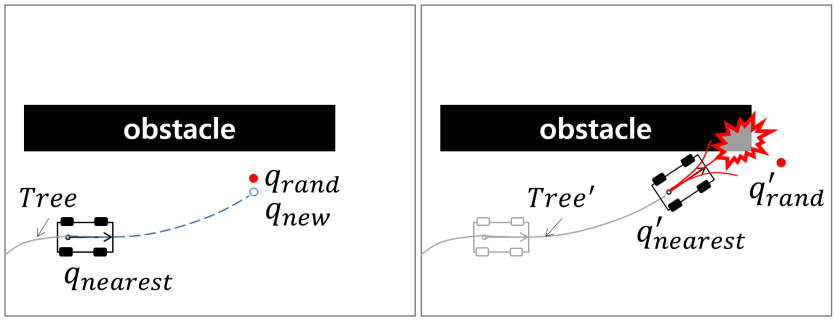
The kinodynamic RRT algorithm is the extend version of the RRT algorithm for nonholonomic systems. The procedure of kinodynamic RRT is as follows. Here, a vehicle's initial and target positions as well as the position of obstacles are all given. A random node (q_{rand}) is selected at each iteration of the algorithm on the configuration space. Then, $q_{nearest}$ becomes the nearest node to q_{rand} among the nodes on a tree topology. The selected $q_{nearest}$ is then expanded toward the q_{rand}. This expanding procedure can be

```
function Kinodynamic-RRT (q_init, q_goal)
τ ← InitializeTree(q_init)
for i = 1 to N do
    q_rand ← RandomSample()
    q_nearest ← FindNearestNeighbor(τ, q_rand)
    u_best ← FindBestControl(q_nearest, q_rand)
    Q_new ← Extend(q_nearest, u_best)
    if CollisionFree(Q_new) then
        τ ← InsertNode(τ, Q_new)
        if q_goal ∈ τ then
            return ExtractTrajectory(τ)
        end
    end
end
return failure
```

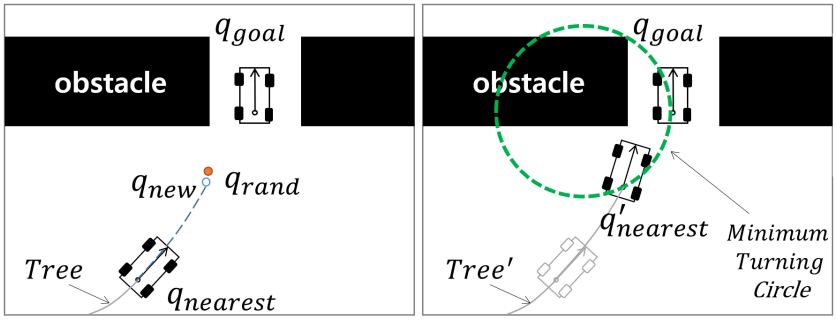
Algorithm 1: Pseudo-code of the kinodynamic RRT algorithm

modeled in a variety of forms depending on vehicle kinematics and environment constraints. Therefore, a control input and q_{new} to be added to the tree are determined. In the extension procedure for a nonholonomic system, a set of candidate paths is generated by considering vehicle dynamics or kinematics. An appropriate candidate (q_{new}) is then selected by using the distance metric to expand from $q_{nearest}$ to q_{rand}. Finally, q_{new} is added to the tree. This process is repeated until the nodes in the tree reach the target point.

The main advantage of this version of the algorithm is that it is useful when considering system characteristics by applying control-based extensions. For this reason, vehicle kinematics and environment model have adopted the control based kinodynamic RRT version to identify a collision-free path. However, this method is less useful in the event that the space is very narrow for a car to pass, or when the accuracy of the positions and orientations is required. This is because the probability is too low that random



(a)



(b)

Figure. 3.1: Illustration of conventional RRT algorithm problems in constrained narrow spaces. (a) the new node cannot be expanded anymore because of obstacles. (b) the new node cannot be converged to the goal position.

nodes are located in a narrow passageway and connected with the desired vehicle posture. Figure.3.1 illustrates these problems in the case of a nonholonomic vehicle system. Figure.3.1(a) is a schematization of nodes that cannot be extended because of surrounding obstacles. The left portion of the picture shows the position of q_{new} reached during the extending process after q_{rand} and $q_{nearest}$ are determined. After a tree is extended from $q_{nearest}$ to q_{new}, a new random sample node q'_{rand} is chosen in the next iteration. However, this node cannot extend to candidate nodes because of obstacles. This

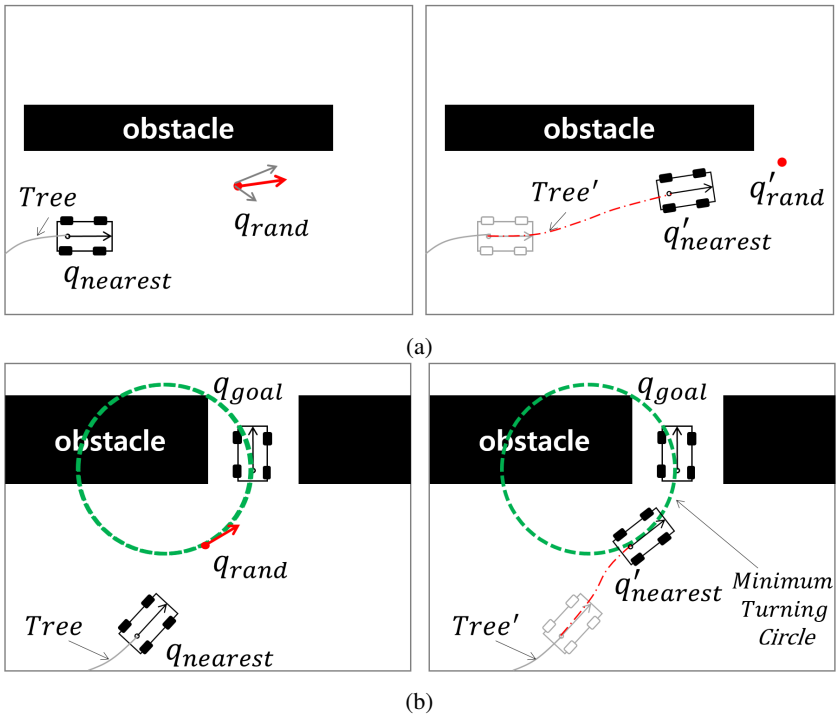


Figure. 3.2: Illustration of DO-RRT algorithm in constrained narrow spaces. The random node is expanded using a desired orientation vector. (a) The direction of random node is generated to avoid collision with obstacles. (b) The desired orientation vector has been determined considering nonholonomic constraints of the vehicle.

can cause considerable computation time and memory usage. Figure.3.1(b) presents the problem that can occur in the case of forward parking, when the desired target position and heading angle are both given. Here, the left portion of Figure.3.1(b) shows the process of selecting q_{new} when considering the relationship between $q_{nearest}$ and q_{rand}. However, the orientation of q_{new} is not suitable for converging on the target position because of the nonholonomic constraints. This can produce a path that requires several repetitions of forward and backward movements in order to reach the target position. To

solve this problem, a novel method assigns a desired orientation for random samples that considers nonholonomic constraints and obstacle positions, as shown in Figure.3.2. Our proposed algorithm is described in detail in the following subsections.

3.3.1 Overview of DO-RRT Algorithm

The DO-RRT algorithm models nonholonomic constraints of the vehicle and geometric constraints of obstacles on the tangential vector space and extends a tree to consider the desired orientations of random samples. Pseudo code is shown in Algorithm 2. This algorithm has four main phases: generating the random node (RandomSampling), finding a nearest node (FindNearestNeighbor), assigning the desired orientation of a random node (GetDesiredOrientation), and extending the tree (DO Extend). The third and fourth phases represent contributions of the proposed algorithm. The third phase determines the desired orientation of a random node using a magnetic-field-based model. The fourth phase generates trajectories by considering both the orientation of the near node and the desired orientation of the random node. Further details are given in next sections.

3.4 Magnetic-like Field based Desired Orientation Model

A magnetic field is a region in space where a magnetic force can be detected. This field can be described using vectors that indicate the magnitude and direction of magnetic forces. The shape of a magnetic dipole field is

function DO_RRT $(q_{init}, q_{goal}, Q_{obs})$
$\tau \leftarrow$InitializeTree(q_{init})
for $i = 1$ **to** N **do**
  while $\rho > \rho_{max}$ **do**
    $q_{rand} \leftarrow$*RandomSample()*
    $q_{rand}, \theta_{pot}, \rho \leftarrow$*PotentialField*$(q_{rand})$
  end
  $\vec{V}_{obs} = \frac{\rho}{\rho_{max}}\vec{V}_t(\theta_{pot}) + (1 - \frac{\rho}{\rho_{max}})\vec{V}(\theta_{rand})$
  $q_{nearest} \leftarrow$*FindNearestNeighbor*(τ, q_{rand})
  $\theta_{mag}, \beta \leftarrow$*MagneticField*$(q_{nearest}, q_{rand}, q_{goal})$
  $\vec{V}_{do} \leftarrow \beta\vec{V}(\theta_{mag}) + (1 - \beta)\vec{V}_{obs}$
  $q'_{rand} \leftarrow$*UpdateOrientation*$(q_{rand}, \vec{V}_{do})$
  $u_{best} \leftarrow$*FindBestControl*$(q_{nearest}, q'_{rand})$
  $Q_{new} \leftarrow$ *DO_Extend*(q_{near}, u_{best})
  if *CollisionFree*(Q_{new}) **then**
    $\tau \leftarrow$ *InsertNode*(τ, Q_{new})
    if $q_{goal} \in \tau$ **then**
      return *ExtractTrajectory*(τ)
    end
  end
end
return *failure*

Algorithm 2: Pseudo-code of the DO-RRT algorithm

similar to that of nonholonomic constraints of the vehicle model. A dipole-like potential energy model is proposed for nonholonomic robots [46]. Another characteristic of magnetic fields exists: the direction of magnetic fields is from the N pole of one magnet to the S pole of the other magnet when dipole moments of two magnetics occur within an influence space. The left portion of Figure.3.3 shows the shape of two magnetic dipole fields (using arrows). The direction of fields is shown from the N pole of the lower magnet to the S pole of the upper magnet. This is similar to the shape of Dubins' path, which is formed by connecting tangential circular arcs and a

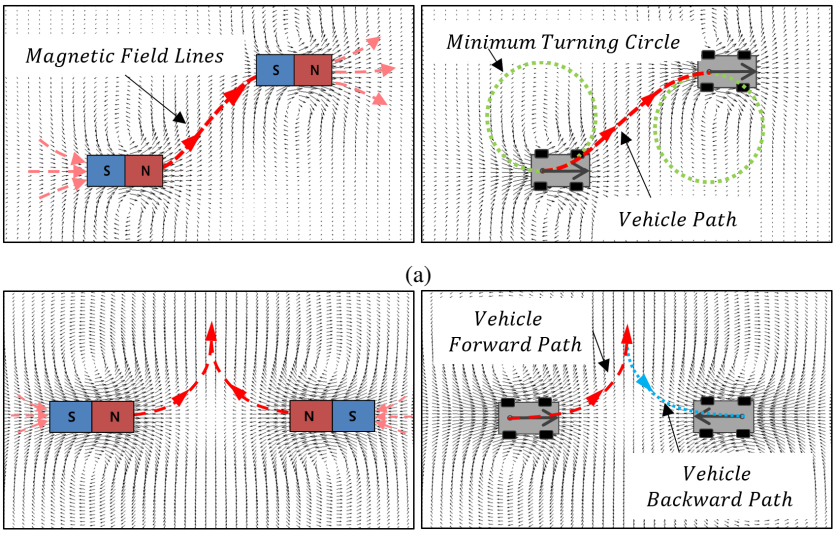


Figure. 3.3: Illustration of the relationship between magnetic field lines of two magnets and vehicle paths. (a) Direction of fields has similar charateristics to the shape of Dubins path. (b) Both forward and backward paths can be determined using magnetic field lines

straight line [47]. By utilizing this characteristic, a vehicle's orientation can be determined to consider its nonholonomic constraint of both forward and backward motions. This can be applied equally in a situation in which an initial position faces the target position. In this section, the magnet-field-based model similar to the nonholonomic constraints of the vehicle is proposed to determine desired orientations of random nodes for sampling-based approaches.

3.4.1 Magnet-like Field Model

Magnetic fields can be modeled by utilizing the Biot-Savart law [48]. This model provides the magnitude of a magnetic field in terms of the elec-

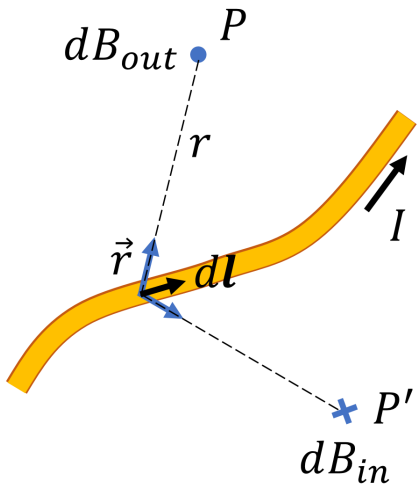


Figure. 3.4: The magnetic field dB at point P and $P^{'}$ is computed by the Biot-Savart law.

tric current. When the currents run in opposite directions on two straight wires, the magnetic dipole vector field can be modeled by applying the superposition principle of magnetic fields as follows.

$$\overrightarrow{B}(q) = K \int_c \frac{\overrightarrow{dl}}{|\overrightarrow{r}|^2} \times \frac{\overrightarrow{r}}{|\overrightarrow{r}|^2} \tag{3.5}$$

where $\overrightarrow{dl}$ refers to the current-carrying segment vector parallel to the Z-axis and $\overrightarrow{r}$ indicates the distance vector between the current-carrying segment and the position q. K is the magnetic constant. The similarity between the magnetic vector field and nonholonomic constraints of bicycle models can be verified as follows. Let us define the current-carrying segment vector $\overrightarrow{dl}$ parallel to the Z axis, and d as the distance between two wires. Here, locations of two wires crossing on the xy surface are derived as

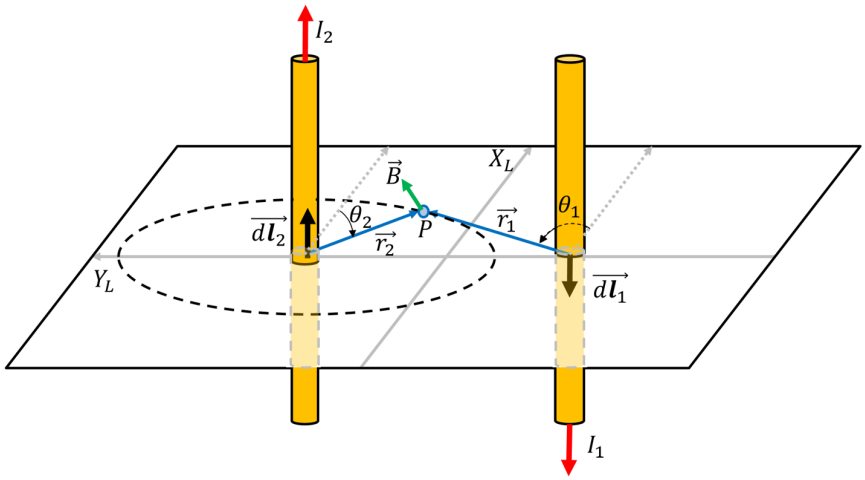


Figure. 3.5: The magnetic field between two wires

$(-\frac{d}{2}\sin(\theta)\hat{i}+\frac{d}{2}\cos(\theta)\hat{j})$ and $(\frac{d}{2}\sin(\theta)\hat{i}-\frac{d}{2}\cos(\theta)\hat{j})$, respectively. Then the magnetic field can be expressed as

$$\vec{B}(q) = K\int\left(\frac{\|\overrightarrow{dl_1}\|}{|\vec{r_1}|^2}(\sin\theta_1\hat{i}-\cos\theta_1\hat{j})+\frac{\|\overrightarrow{dl_2}\|}{|\vec{r_2}|^2}(-\sin\theta_2\hat{i}+\cos\theta_2\hat{j})\right). \tag{3.6}$$

3.4.2 Pfaffian Constraints

In this section, Pfaffian constraints of the vehicle are briefly described. First we consider a kinematic model of the vehicle with front and rear wheels. For simplicity, we use a bicycle model with a pair of front wheels and rear wheels as single wheels in the center of x-axis. The rear tire is aligned with the x-axis of the vehicle, and the front tire can rotate around the z-axis. It is assumed that there is no slip of the wheel. Let x_G, y_G, and

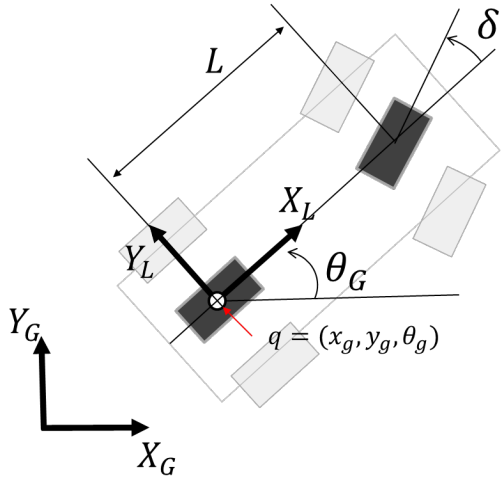


Figure. 3.6: Kinematic model of the vehicle

θ_G be the position and heading angle with respect to the center axis of the rear wheel and δ is the steering of the car as shown in Figure.3.6. The non-holonomic constraints for the rear wheel can be derived by setting the wheel side slip speed to zero. The Pfaffian constraint of the rear wheel can be expressed as:

$$A^T(q)\dot{q} = \begin{pmatrix} \sin(\theta_g) & -\cos(\theta_g) & 0 \end{pmatrix} \dot{q} = 0. \tag{3.7}$$

By substituting $\dot{q} = (\dot{x}_g, \dot{y}_g, \dot{\theta}_g)$ for the direction vector $\overrightarrow{B(q)}$ of the magnetic field, it can be proved that the non-holonomic constraint is satisfied as follow.

$$A^T(q)\overrightarrow{B}(q) = K_1 \cos\theta_1 - K_2 \cos\theta_2 \tag{3.8}$$

where

$$\overrightarrow{B(q)} = \begin{pmatrix} K_1 \sin(\theta_1 + \theta_g) - K_2 \sin(\theta_2 + \theta_g) \\ -K_1 \cos(\theta_1 + \theta_g) + K_2 \cos(\theta_2 + \theta_g) \\ 0 \end{pmatrix}, K_i = K \int \frac{\|\overrightarrow{dl_1}\|}{|\overrightarrow{r_1}|^2}.$$

Here, when q is set to (x_g, y_g, θ_g), θ_1 and θ_2 are $\frac{\pi}{2}$ and $-\frac{\pi}{2}$, respectively. Therefore $A^T(q)\overrightarrow{B}(q) = K_1 \cos(\frac{\pi}{2}) - K_2 \cos(-\frac{\pi}{2})$ is 0. This is satisfied as Pfaffian constraint of the rear wheel. That is, the magnetic field vector converges to the origin parallel to the x-axis as it approaches the origin.

3.4.3 DO(Desired Orientation) Model

The magnetic vector field and nonholonomic constraints of bicycle models have similar characteristics. Using this relationship between the magnetic vector field and nonholonomic constraints, a magnetic-field-based model for the initial and target positions of the vehicle is derived as follows.

$$\begin{aligned} \overrightarrow{V_m}(q) = k_m \Bigg[\int_{c_s} \left(\frac{\overrightarrow{dl_{s_1}}}{|\overrightarrow{r_s}|^2} \times \frac{\overrightarrow{r}}{|\overrightarrow{r}|^2} + \frac{\overrightarrow{dl_{s_2}}}{|\overrightarrow{r_s}|^2} \times \frac{\overrightarrow{r}}{|\overrightarrow{r}|^2} \right) + \\ \int_{c_g} \left(\frac{\overrightarrow{dl_{g_1}}}{|\overrightarrow{r_g}|^2} \times \frac{\overrightarrow{r}}{|\overrightarrow{r}|^2} + \frac{\overrightarrow{dl_{g_2}}}{|\overrightarrow{r_g}|^2} \times \frac{\overrightarrow{r}}{|\overrightarrow{r}|^2} \right) \Bigg] \end{aligned} \tag{3.9}$$

where $\overrightarrow{r_s}$ is the directional vector between the current-carrying segment and the vehicle's start position and $\overrightarrow{r_g}$ is the target position. Here, the distance between two wires is equal to the width of a vehicle. For the convergence at the initial or target position of the vehicle, desired orientation vectors are calculated to satisfy the nonholonomic constraint. Moreover, the desired orientation vector between two positions can be computed to reach the other position based on the forward or backward movements of the vehicle.

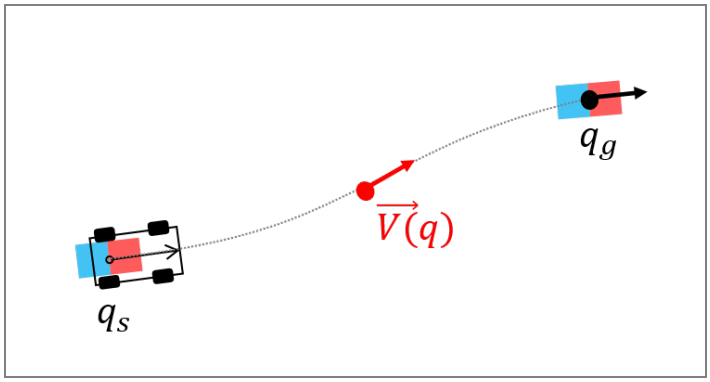


Figure. 3.7: Desired orientation vector based on the magnetic model

In Figure.3.8, the validity of the magnetic vector direction was verified by generating a path between the initial position and the target position. Given the direction of the magnetic vector, it can be seen that both forward and backward paths as well as the forward path are useful in connecting the two points. This method has the advantage that the feasible direction can be calculated independently from other regions. Due to its low computational complexity, it is also suitable for applying to sampling-based path generation methods. Figure.3.9 is a schematic representation of the results of considering the vehicle constraints at the sampling stage of the RRT algorithm. Considering non-holonomic constraints based on magnetic field, it not only increases the number of nodes in an expandable direction, but also guides the convergence direction to the target point.

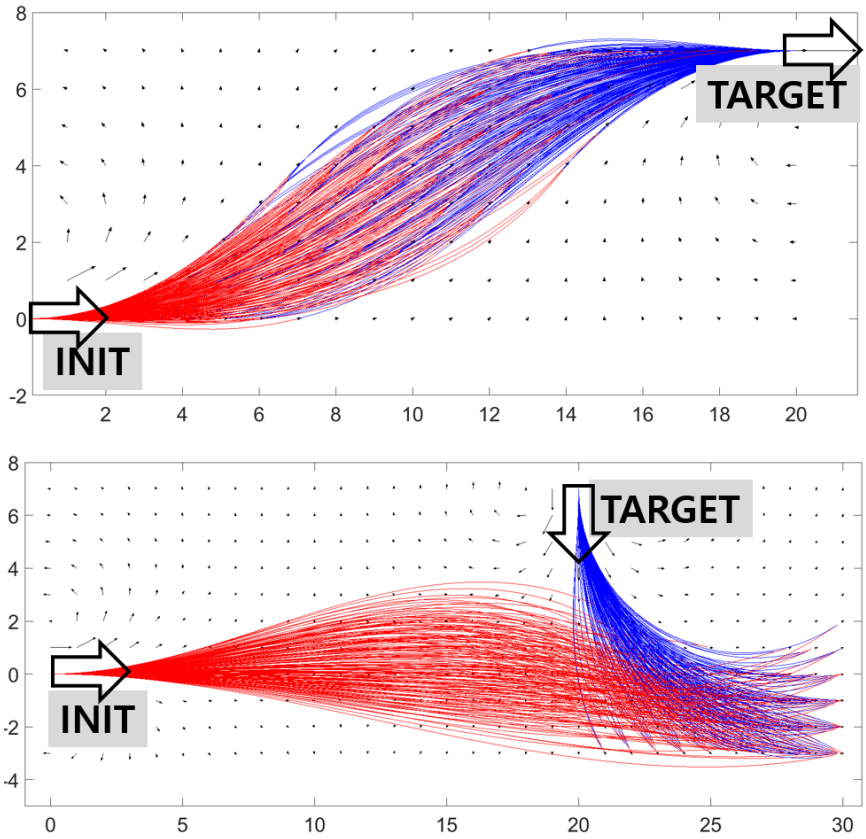


Figure. 3.8: Feasibility tests of desired orientation model

3.5 Sampling Fuction of DO-RRT

In the DO-RRT Algorithm, the desired orientation of a vehicle at the q_{rand}, as explained later, can be determined by utilizing the magnetic-field-based model on q_{near} and q_{goal}. However, adopting the magnetic-field-based model is difficult when the distance between $q_{nearest}$ and q_{goal} is too far to influence magnetic forces. In addition, the direction of magnetic field lines

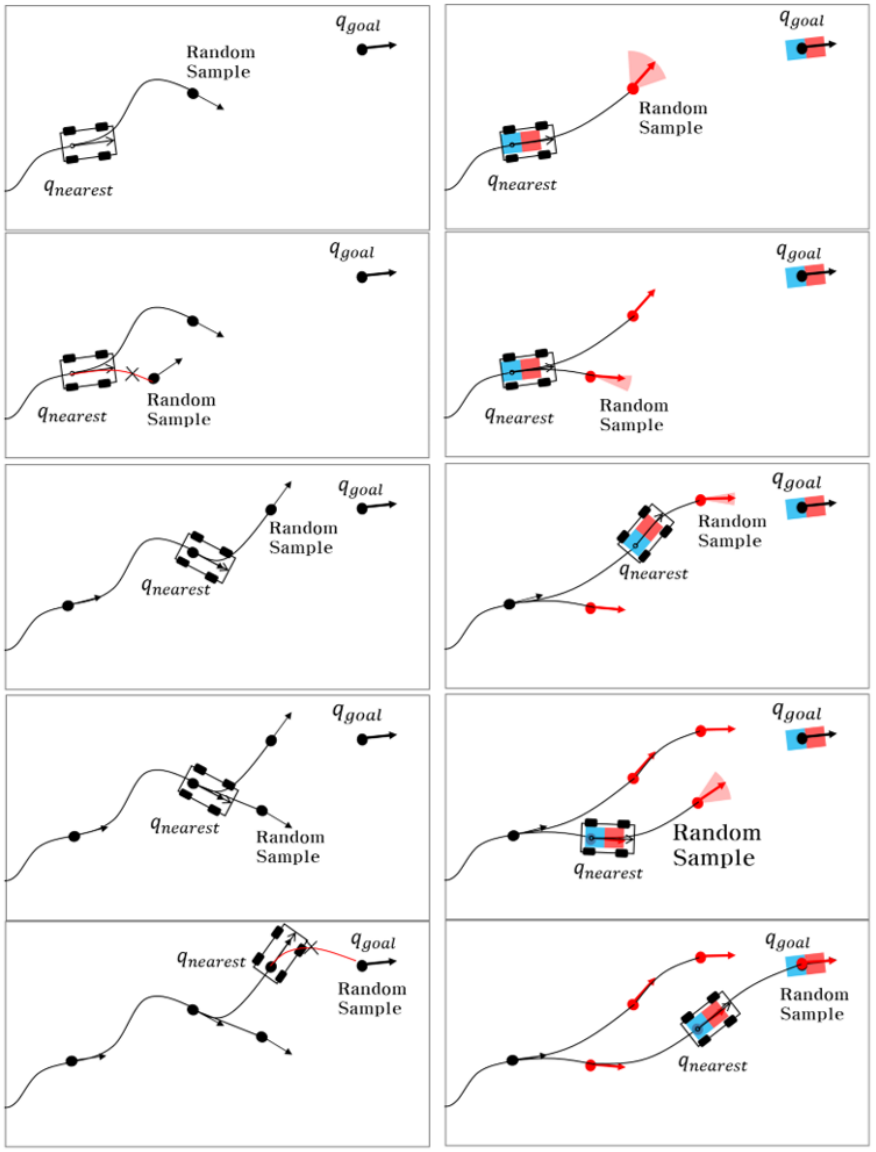


Figure. 3.9: Comparison of sampling strategies. (Left: Goal biased sampling method, Right: Magnetic field based sampling method)

may not be valid depending on the presence of obstacles. For these reasons, the desired orientation can be modeled as the sum of three type of vectors as follows.

$$\overrightarrow{v} = k_m\overrightarrow{V_m}(q) + k_r\overrightarrow{V_r}(q) + k_o\overrightarrow{V_o}(q), \|\overrightarrow{V_m}(q)\| < \varepsilon \tag{3.10}$$

where $\overrightarrow{V_r}$ is the random direction vector weighted to the direction of the target position and $\overrightarrow{V_o}$ is the direction of the repulsive force from the obstacle. The value of k_r gain is determined by considering the distance between q_{node} and q_{near}, or q_{goal}. The k_m and the k_o are set to the magnitude of the magnetic and obstacles' potentials, respectively. In case of a long distance of q_{rand} from $q_{nearest}$ and q_{goal}, k_r gain is set to be greatly affected by the random direction. The desired orientation vector is valid when the magnitude of $\overrightarrow{V_m}(q)$ is less than the threshold value (ε). To avoid selecting an invalid random node, the value of ε is assigned by considering the correlation between the magnitude of $\overrightarrow{V_m}(q)$ and a minimum turning radius.

3.6 Extend Function of DO-RRT

The proposed extend function of DO-RRT generates trajectories using the desired orientation and position of q_{near} and q_{rand}. First, candidate trajectories are calculated for the set of velocities for forward and backward driving. For the set of determined velocities, the steering angle is computed to converge to the target position using the SteeringController function. Using this steering angle, a ForwardSimulation function calculates the estimated vehicle positions after Tsec. This process is repeated N times until

```
function DO_Extend(q_near, q_rand)
for i = 1 to Size(V) do
    v_d = V(i)
    q_0 ← q_near
    for j = 1 to N do
        δ_j ← SteeringController(q_0, q_rand, v_d)
        Q_cand ← ForwardSimulation(Q_cand, q_0, δ_j, v_d, T)
    end
    Q_new ← CostFuction(Q_cand)
end
return Q_new
```

Algorithm 3: Pseudo-code of the expand function for the DO-RRT algorithm

a candidate trajectory (Q_{cand}) is generated which then converges to a target position. These candidate trajectories with respect to the set of velocities are compared to select a best path (Q_{new}) using the cost function. This cost function can be derived using the position and orientation error between q_{new} and q_{rand} and number of changes in the direction.

3.7 Experimental Results

In this section, the DO-RRT algorithm is analyzed in a narrow parking space using simulation and autonomous vehicle test.

3.7.1 Experimental Condition

All Experiments were carried out in a C++ implementation on Intel (R) Core (TM) i5-2500 CPU @ 3.30GHz CPU and 4G Memory and Ubuntu 14.04.3 LTS operating system. Vehicle-specific parameters for the path planning were chosen based on an autonomous vehicle in our labora-

tory (SPIRIT-1, Hyundai Grandeur). Vehicle length and width are 2845 and 1614 mm, respectively. The distance from the center of the vehicle to the real wheel center is 1100 mm and to the front wheel center is 1745 mm. The maximum wheel angle is 35.0°.

3.7.2 Simulation Test Results

Experiments involving forward, backward, and parallel parking were conducted to compare the performances of the DO-RRT. These parking simulations were performed in narrow spaces that required driving both forward and backward. In addition, the experiments were conducted to generate a path through the environment without obstacles. We also applied a standard RRT-based nonholonomic planner within simulations for comparisons in the proposed algorithm. Here, this nonholonomic planner could be modified according to the type of robot (kinematic model, with kinematics and dynamics constraints, motion direction, etc.) and workspaces (dimension of the configuration space, shape of obstacles, etc.). A parking path was generated in the narrow space by implementing a nonholonomic version of the RRT algorithm to allow forward and backward movements based on car-like robots. Both the DO-RRT and nonholonomic RRT adopted a goal-biased sampling method and a same distance metric function to identify nearest nodes. Unlike the holonomic system, the nonholonomic RRT-based method had a significant effect on the metric's performance. The proposed method calculates the orientation of the valid direction from random samples; this process can reduce the number of non-scalable nodes due to incorrect metrics. Therefore, the proposed method computes the distance through a sim-

ple weighted Euclidean metric as follows.

$$Dist(q,q') = \int_0^T l(q(t),u(t))dt + w_p \parallel p - p' \parallel^2 + w_q \parallel \theta - \theta' \parallel^2 + w_v \parallel v - v' \parallel \quad (3.11)$$

where w_p, w_q, and w_v are weight values, p is a position vector, θ is an orientation value, and v is the linear velocity at state q. The collision detection used in this study was implemented using the k-d tree. The k-d tree refers to a space-division data structure for constructing points in a k-dimensional space that extends a binary search tree into a multidimensional search space. Collisions with obstacles are determined by measuring the number of nodes of the k-d tree that existing within a safe distance d with respect to an arbitrary point p. Figure.3.10 shows details of the experiments; the proposed DO-RRT algorithm was compared to the nonholonomic RRT method using the iteration count and path length to reach the target position. These performances were measured using the average and standard deviation of 100 trials. According to the paths generated from Cases $1-4$, the number of iterations of the DO-RRT proved much fewer than those of the nonholonomic-RRT. The path length using the DO-RRT was also shorter than that using nonholonomic-RRT. The path from the nonholonomic RRT has many direction changes that reduce the heading error in the narrow region. Therefore, the length of the path increased. Overall, the performance was improved, as it required 9.402 times fewer iterations on average in four cases and the path length was decreased to about 0.834 of its original length.

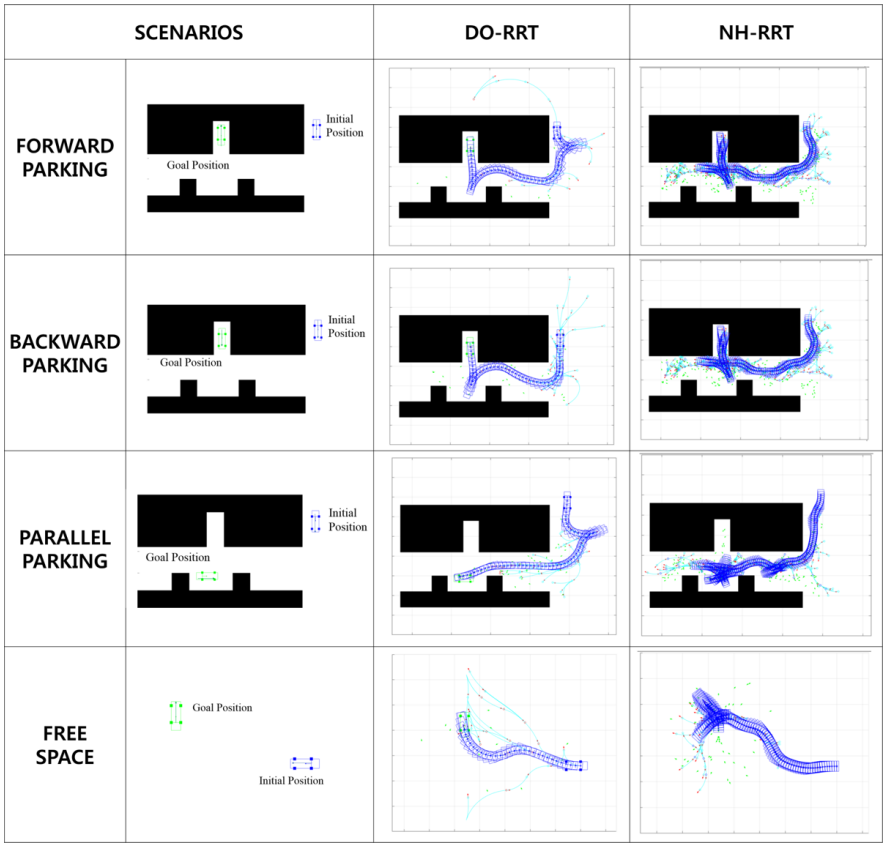


Figure. 3.10: The environments of four parking cases and example solutions found with the DO-RRT and Nonholonomic RRT (NH-RRT).

3.7.3 Vehicle Test Results

Experiments were conducted to generate a parking path using the autonomous vehicle system shown in Figure.3.11. The environmental information was collected from the 3D lidar (Velodyne lidar). The ego vehicle position and heading angle were estimated by the simultaneous localization and mapping (SLAM) based on Velodyne and inertial measurement unit

Table 1: Comparison Results between DO-RRT and Conventional Nonholonomic RRT

		# of iteration		Length(m)	
		Average	Std.Dev	Average	Std.Dev
CASE1	DO-RRT	31.328	14.190	49.925	5.919
	NH-RRT	131.500	63.986	54.827	3.250
CASE2	DO-RRT	23.289	9.249	49.768	4.739
	NH-RRT	298.637	191.035	61.795	13.574
CASE3	DO-RRT	44.908	53.033	48.066	10.606
	NH-RRT	367.400	275.129	73.018	7.611
CASE4	DO-RRT	7.666	2.546	41.166	3.562
	NH-RRT	95.139	64.972	42.693	7.532

*DO-RRT: Desired Orientation RRT, *NH-RRT: Nonholonomic RRT

(IMU) data. The test procedure is described as follows. When an available parking space was identified by a user, an autonomous navigation system drove the car to that parking space by following waypoints. When the vehicle reaches the parking space, a parking path is computed from the current position of the vehicle to the parking spot using the DO-RRT algorithm. Experimental results are shown in Figure.3.12. The upper figure shows the point cloud map obtained by the Velodyne sensor in the parking lot where the experiment was conducted. The target position of the vehicle required backward parking. Experiments were conducted from two initial positions. In the first test, each of forward and backward movement were performed without collision with obstacles. The second situation was difficult to conduct directly because of other parked vehicles. In this case, the parking path for the backward parking was generated using two backward and one forward movements.

(a)

Sensors		Characteristics and configurations
ANALOG DEVICES ADIS 16405	Gyroscope	• 3D rate of turn with 0.007 degree/s bias stability • 100Hz output rate
	Accelerometer	• 3D acceleration with 3.33 mg resolution • 100 Hz output rate
Wheel speed sensors		• 50 Hz output rate
Velodyne 64-E		• 64 Channels Lidar • <2cm accuracy • 0.08° angular resolution • 5~15Hz output rate

(b)

Figure. 3.11: The experimental autonomous vehicle system : (a) SPIRIT-1, (b) Specification of sensors

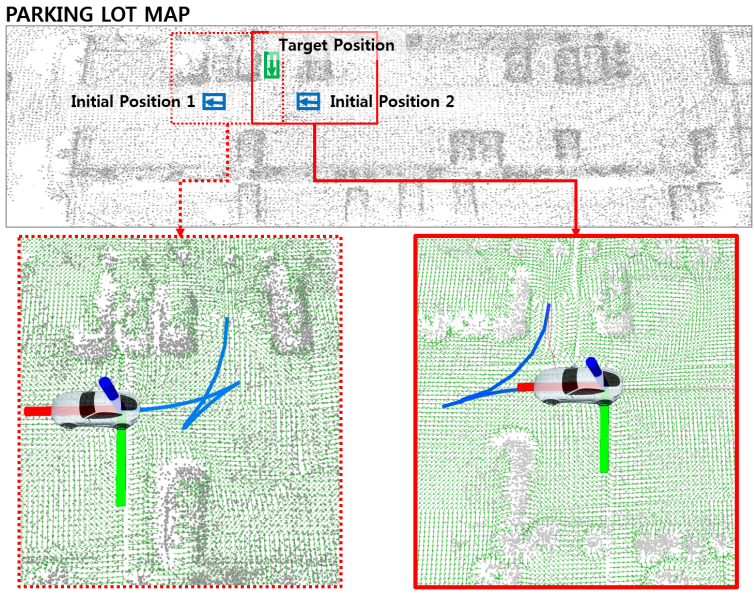


Figure. 3.12: Results of trajectories generated by DO-RRT algorithm in a standard parking lot.

Chapter. 4

Sampling-based Geometric Motion Planning Algorithm for Narrow Cluttered Environments

4.1 Overview

This chapter covers robust sampling-based nonholonomic motion planning methods in narrow and cluttered areas. The control-based kinodynamic method presented in Chapter 3 does not use a steering method but instead uses distance metric based functions [19]. This makes the RRT unable to efficiently explore the state space. In addition, the control-based kinodynamic RRT algorithm cannot generally guarantee probabilistic completeness [49]. Unlike the control-based RRT, the geometric-based RRT method relies on the interpolation method rather than the state propagation method. Therefore, the proposed algorithm is developed using the geometric-based method to determine the optimal solution. Contributions are as follows; first, in narrow and cluttered areas, the probability of obtaining a sample that can pass through an area cluttered with obstacles is relatively low compared to one that can pass through an open area. It may also fail to extend the path if the sampled position is difficult to extend from adjacent nodes. A constraint model on the tangential direction of the random sample is proposed

to solve this problem. Second, we propose an extension method based on the tangential direction constraint. In the process of expanding the tree to random samples, many nodes in narrow and cluttered regions cannot pass the collision test. This increases both the number of unnecessary iterations and memory usage. We propose a node extension method that is based on gradient descent to solve this problem.

4.2 Backgrounds

In this section, we describe the geometric-based version of the nonholonomic RRT and RRT* algorithms and the problems that arise in a narrow area such as a parking space.

4.2.1 Algorithm Description and Limitations

The RRT algorithm searches the path by expanding the tree until it arrives at the target position. Here, a vehicles initial and target positions as well as the position of obstacles are all given. A random node (q_{rand}) is selected at each iteration of the algorithm on the configuration space. Then, the node closest to the tree node ($q_{nearest}$) is searched between nodes on the tree topology. This nearest node expands toward the random node within the reachable set in the expansion procedure. This reachable set can be modeled in a variety of forms depending on vehicle kinematics and environment constraints. Finally, a new node (q_{new}) to be added to the tree is determined. This procedure is repeated until the target position is reached. Pseudo code is shown in Algorithm 4. The probabilistic completeness of this RRT algo-

```
function RRT (q_init, q_goal)
τ ← InitializeTree(q_init)
for i = 1 to N do
    q_rand ← RandomSample()
    q_nearest ← FindNearestNeighbor(τ, q_rand)
    q_new ← Extend(q_nearest, q_rand)
    if CollisionFree(q_new) then
        τ ← InsertNode(τ, q_new)
        if q_goal ∈ τ then
            return ExtractTrajectory(τ)
        end
    end
end
return failure
```

Algorithm 4: Pseudo-code of a geometric version of the RRT algorithm

rithm has been proved. In other words, as the sampling number increases infinitely, the probability of finding a solution converges to 1. However, this method cannot guarantee the optimal path.

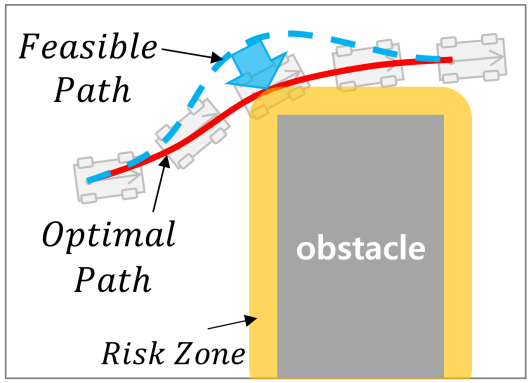
RRT* algorithm is proposed as an extended version of RRT for optimal path planning problems. Unlike the RRT method, RRT* algorithm is improved to maintain the lowest cost node connectivity when nodes are added. RRT* pseudo code is described in Algorithm 5. The RRT* algorithm has two additional procedures for optimizing the RRT algorithm. The first step is to find the lowest node (q_{min}) by comparing q_{new} with the cost of nodes within a specified expanding range. The node expansion is performed in the direction of q_{min} to q_{new}. Here, the specified expanding range is determined by $\gamma(\log(n)/n)1/d$ to ensure asymptotic optimality. γ is a constant related to the volume of the search space, n is the number of iterations, and d is the dimension of the configuration space. This expanding range tends to

```
function RRT* (q_init, q_goal, M)
τ ← InitializeTree(q_init)
for i = 1 to N do
    q_rand ← RandomSample()
    q_nearest ← FindNearestNode(τ, q_rand)
    q_new ← Extend(q_nearest, q_rand)
    if CollisionFree(q_new) then
        Q_near ← FindNearNode(τ, q_new, γ)
        q_min, isValid ← ChooseParent(q_new, Q_near)
        if isValid then
            τ ← InsertNode(τ, q_min, q_new)
            τ ← Rewire(τ, q_new, Q_near)
            if q_goal ∈ τ then
                return ExtractTrajectory(τ)
            end
        end
    end
end
return failure
```

Algorithm 5: Pseudo-code of the RRT* algorithm

decrease gradually as the iteration progresses. The second procedure compares expansion costs from the newly added node (q_{new}) to neighbor nodes (Q_{near}), and rewires for the lower cost node than the previous one. This rewire process ensures optimal connection of nodes. Figure.4.1 illustrates the convergence of the feasible solution to the optimal solution as iteration increases infinitely.

The major advantage of this algorithm is that it is useful for considering system characteristics. For this reason, geometric-based RRT and RRT* algorithms have been adopted to identify collision-free paths to nonholonomic vehicles. However, these method are less useful when the space is too narrow to pass the vehicle, or when the accuracy of the positions and



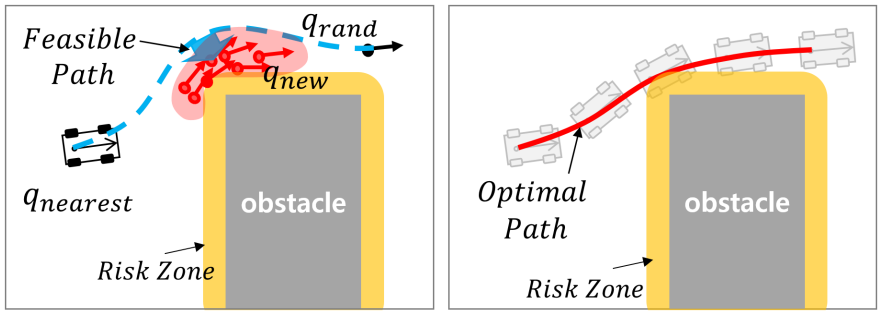
(a)

Figure. 4.1: Convergence process to optimal path with increasing iteration

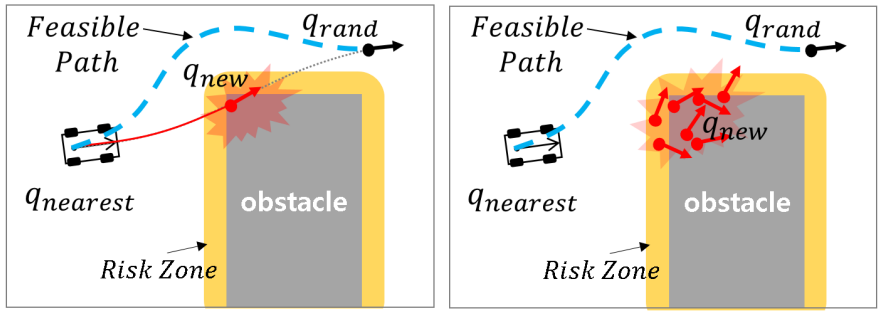
orientations is required such as a parking lot. This is because the probability is too low that random nodes are located in a narrow passageway and connected with the desired vehicle posture. This is identical to problems described in Chapter 4. In addition, the RRT* algorithm which computes the optimal solution, can cause convergence problems in constrained narrow spaces. Figure.4.2 illustrates RRT* problems of a nonholonomic vehicle system.

4.2.2 Overview of Proposed Algorithm

A proposed algorithm models nonholonomic constraints of the vehicle and geometric constraints of obstacles on the tangential vector space and extends a tree to consider the desired orientations of random samples. Desired Orientation based Sampling (DO-Sampling) and Extension (DO-Extension) are contributions of the proposed algorithm. These methods are applied by replacing the random sampling function and the extend function of the RRT



(a)



(b)

Figure. 4.2: Illustration of the conventional RRT* algorithm problem in constrained narrow spaces. (a) Node expansion near the obstacle plays an important role in ensuring that the feasible path converges to the optimal path. (b) Nodes near obstacles have less possibility to expand due to collisions. In this reason, a large number of iterations are required to converge to the optimal path.

and RRT* algorithm, respectively. In the DO-Sampling phase, the desired orientation of a random node is determined by using a magnetic-like field model. The DO-Extension phase generates trajectories by considering both the orientation of the near node and the desired orientation of the random node. In particular, to improve convergence rate of asymptotic optimality, a method for determining q_{new} nodes based on gradient descent is proposed. Further details are given in the next section.

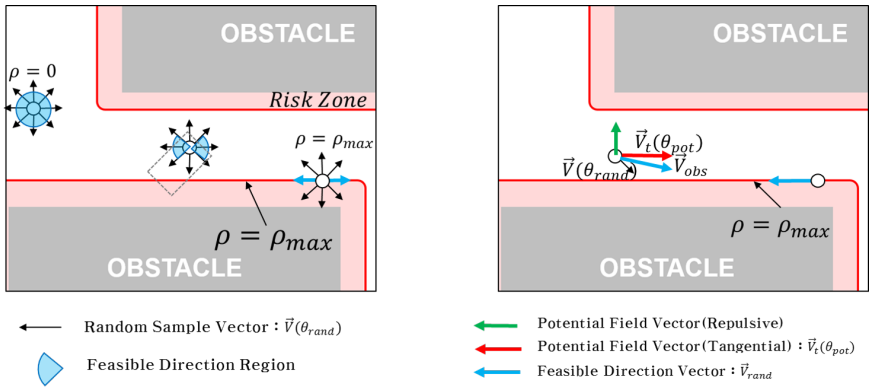


Figure. 4.3: Determination of feasible regions based on the potential field

4.3 Desired Orientation based Random Sampling Method

In this section, we propose a sampling method considering the constraints of the environment and the system constraints of the vehicle. The nonholonomic constraints of the vehicle are determined using magnetic-like field based models. This model provides information on the reachable direction from the start position, and the convergence direction to the target position. The direction of the constraint is determined by the magnetic field, which is the magnitude of the magnetic field. In a region with a large magnetic force, the direction of the magnetic field should be selected as a random sample in order to satisfy the non-holonomic constraint. Therefore, in the space where the obstacle does not exist, RRT planner can quickly find a path for converging to the target position by applying the direction of the magnetic field to the random sample. In a narrow region, the probability of

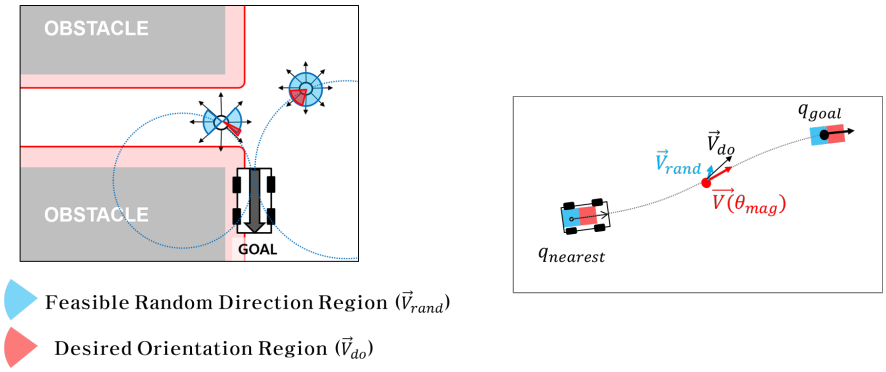


Figure. 4.4: Desired orientation based random sampling method

random samples assigning to feasible regions is low due to the obstacles. Therefore, it is important to increase the number of samples that can be expanded considering the constraints of obstacles. To solve this problem, the potential field is applied to the constraint model of the sampling method. First, the potential field is applied to obtain a random sample in the free space (without obstacles). The direction of the sample is also determined using the potential field so that the sample around the obstacle is in an expandable direction. Pseudo code is described in Algorithm 6.

RandomSample is a function that generates samples on the configuration space. Both uniform sampling and goal-biased sampling strategies can be used as *RandomSample* function. To determine the validity of the sample, the magnitude and direction of the potential field at the random sample(q_{rand}) is obtained via the *PotentialField* function. This process is repeated until the magnitude of the potential field is lower than a threshold value(ρ_{max}). Random samples generated within the collision area can be removed through this step. Nodes that are generated near obstacles have dif-

function DO-RandomSample(q_{init}, q_{goal})
while $\rho > \rho_{max}$ **do**
  $q_{rand} \leftarrow$ *RandomSample()*
  $\theta_{pot}, \rho \leftarrow$ *PotentialField*(q_{rand})
end
$\vec{V}_{rand} = \frac{\rho}{\rho_{max}}\vec{V}(\theta_{pot}) + (1 - \frac{\rho}{\rho_{max}})\vec{V}(\theta_{rand})$
$\theta_{mag}, \beta \leftarrow$ *MagneticField*$(q_{init}, q_{rand}, q_{goal})$
$\vec{V}_{do} \leftarrow \beta\vec{V}(\theta_{mag}) + (1 - \beta)\vec{V}_{rand}$
$q_{rand} \leftarrow$ *UpdateOrientation*$(q_{rand}, \vec{V}_{do})$
return q_{rand}

Algorithm 6: Pseudo-code of the sampling function for the DO-RRT/RRT* algorithm

ferent extendability depending on the direction. For this reason, not only the position of the random node but also the direction of the random node have a significant influence on the performance of the RRT algorithm. A feasible direction of the sample is calculated using the relationship between the tangential vector of the random sample and the tangential vector of the potential field. The weighting factor of each tangential vectors is assigned by using the magnitude of the potential energy. In the region where the potential energy is high, the direction of the sample is positioned in the direction of low energy. Conversely, in regions with low potential energy, the direction of sample is predominantly assigned to the direction of the random vector. The nonholonomic constraint of the vehicle can also be formulated in this way through the magnetic-like field. The magnetic field represents the direction of the vehicle that satisfies the non-holonomic constraint. That is, as the magnetic force increases, the direction of the magnetic field coincides with the reachable direction of the vehicle. This information is calculated by the *MagneticField* function. θ_{mag} is the direction angle of the magnetic

field, and β means the magnitude of the magnetic field and is nominalized between 0 and 1. Finally, the angle of the determined desired orientation sample is updated via the *UpdateOrientation* function.

4.4 Desired Orientation based Extend Method

function DO-Extend($q_{nearest}$,q_{rand})
$q_{new} \leftarrow$Extend($q_{nearest}$,q_{rand})
for $i = 1$ **to** N_{max} **do**
  θ_{pot},$\rho \leftarrow$*PotentialField*(q_{new})
  $q_{new} \leftarrow q_{new} + \lambda(\rho)\frac{\vec{V}(\theta_{pot})}{\|\vec{V}(\theta_{pot})\|}$
  if *CollisionFree*(q_{new}) **then**
    θ_{mag},$\beta \leftarrow$*MagneticField*($q_{nearest}$,q_{new},q_{rand})
    $q_{new} \leftarrow$*UpdateOrientation*(q_{new},$\vec{V}(\theta_{mag})$)
    break;
  end
end
return q_{new}

Algorithm 7: Pseudo-code of the extend function for the DO-RRT/RRT* algorithm

The Extend procedure generates the vehicle's trajectory from $q_{nearest}$ to q_{rand} taking into account the kinematics or dynamics of the vehicle. Especially in the case of RRT*, the two-point boundary value problem must be solved for rewiring of the nodes. For this reason, this process is the largest part of the computational cost of the RRT algorithm. To increase the performance of the RRT algorithm for nonholonomic systems, it is important to reduce the number of calls to the Extend function. This section discusses a gradient descent based extension method. In the RRT and RRT* algorithms, if the return value of the Extend function, q_{new}, is located in the collision

region, the tree expansion is aborted and the sampling function of the next iteration is performed. Therefore, a large number of extend function calls are required for node expansion in a narrow region. To solve this problem, the q_{new} is shifted in the gradient direction of the potential field so as to avoid the collision with the obstacle. The orientation of the node at the shifted position is then calculated using a magnetic field to satisfy the nonholonomic constraint. Figure.4.5 illustrates this process. The proposed method replaces the *Extend* function of RRT and RRT * with the *DO-Extend* function as shown in Algorithm 7. *DO-Extend* function first calculates q_{new} through the *Extend* function. The potential field information(θ_{pot} and ρ) at the q_{new} position are then obtained in the *PotentialField* function. Using these values, a gradient vector and a small step size(λ) of the gradient descent algorithm are derived. The gradient descent algorithm moves the position of the q_{new} node away from the obstacle. When q_{new} is out of the collision area, the desired orientation angle is computed using the magnetic field. This process is repeated for a predefined number of iterations (N_{max}). The proposed method requires additional computation in the procedure but has two advantages. The first is to find a feasible solution in a narrow environment with only a small amount of sample. The second is that the convergence rate of the optimal solution is improved by increasing the number of valid samples near the obstacle. A detailed analysis of the algorithm is given in Section 4.5.

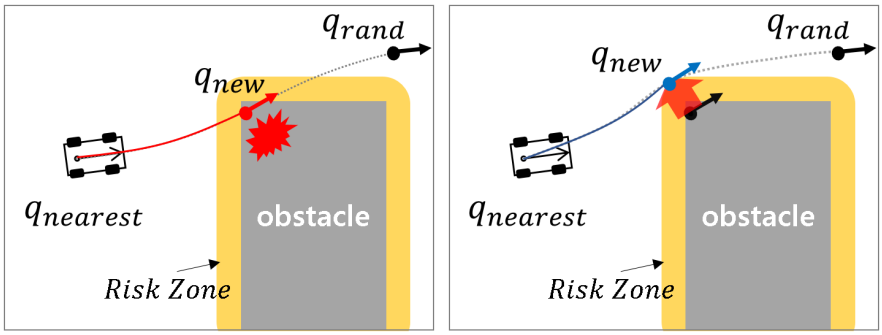


Figure. 4.5: Gradient descent based extend method. The left figure shows a case where q_{new} node for expanding to a random node (q_{rand}) is located in an obstacle area. In this case, the node expansion process is stopped and the next iteration is performed. The figure on the right shows the proposed method, where q_{new} is shifted out of the risk zone in the direction of the gradient vector.

4.5 Analysis

In this section, we analyze the probabilistic completeness and asymptotic optimality of the proposed algorithm. The efficiency of the algorithm is also discussed by analyzing the convergence rate of the proposed algorithm.

4.5.1 Probabilistic Completeness

In the path planning problem, a collision-free path is said to be robustly feasible if it have strong-δ-clearance for $\delta > 0$ [18]. In this problem, the algorithm is said to be probabilistic complete if the probability of finding a solution approaches 1, as the number of iterations of the algorithm increases to infinity. The probabilistic completeness has been proved by referring to the notion of an attraction sequence [19]. The attraction vertex(A_i) is defined as the point that leads the robot to the target position. A finite sequence of

attraction vertices sets from the initial position to the target position is called an attraction sequence $A = \{A_0, A_1, ..., A_n\}$ for $n \in \mathbb{R}_+$. Let A_1 be the initial position(q_{init}) and A_k be the target position(q_{goal}). For each A_i, there exists a basin of attraction B_i. For all $q_{i-1} \in A_{i-1}, q_i \in A_i$, and $q_j \in Q \backslash B_i$ then $d(q_i, q_j) < d(q_{i-1}, q_j)$ where d is the distance metric. For all $q_i \in A_i$ and $q_j \in B_i$, there exists an input sequence $U = \{U_0, U_1, ..., U_m\}$ that carries the state q_j to q_i. Under these conditions, guaranteeing the probabilistic completeness of the RRT algorithm can be derived by expressing the probability that a random sample exists in the Attraction vertex as a Bernoulli distribution. Let $\mu(\cdot)$ denote the measure function on the configuration space.

Theorem 1. *([19]) If an attraction sequence of length k exists, the probability that the RRT algorithm fails to find the path after n iterations is less than* $exp(\frac{-np+2k}{2})$ *where* $p = \min_i \frac{\mu(A_i)}{\mu(Q_{free})}$.

In the DO-RRT algorithm, desired orientated vectors(Q_{do}) is modeled by taking into account the environmental constraints and the nonholonomic constraints of the vehicle. The minimum probability(p) for the DO-RRT algorithm is described in following lemma.

Lemma 1. *In the DO-RRT algorithm, the minimum probability that the random sample lies in the attraction vertex is denoted as* $p' = \min_i \frac{\mu(A_i)}{\mu(Q_{do})}$, *where* Q_{do}=$Q_{free} \backslash Q_{infeaible}$.

The DO-RRT algorithm can reduce the number of random samples in an area not included in the attraction sequence. Thus, the minimum probability(p') that the random sample lies in the attraction vertex is greater than the RRT algorithm. Assumptions for the proof of probabilistic completeness in the DO-RRT algorithm are the same as the RRT algorithm. The probabilistic completeness of the DO-RRT algorithm can be proved as below.

Theorem 2. *If an attraction sequence of length k exists, the probability that the DO-RRT algorithm fails to find the path after n iterations is less than* $exp(\frac{-np'+2k}{2})$.

As the number of random samples goes to infinity, the probability of failure to find a solution is smaller than that of RRT through the relationship of $p \leq p'$. In other words, the DO-RRT can find a solution with a small number of random samples.

4.5.2 Asymptotic Optimality

The RRT* algorithm ensures the asymptotic optimality as the number of samples increases infinitely. In [50], it has also been proven that the differential constrained version of the RRT* algorithm guarantees the asymptotic optimality if the following two conditions hold: 1) The system must satisfy the weakened local controllability and 2) The optimal path must satisfy ε-collision-free. It should also be defined a circle with the center at q_2 that is reachable from q_1. This ensures that q_1 and q_2 can be connected when the

sampling inside this circle is infinite. In condition 1), a basin of attraction (ε-reachable set) must be nonempty interior. The nonholonomic system such as a car-like model satisfy the weakened local controllability assumption [50]. The condition 2) is related to the search range of a near node. In [18], it has been proved that assumption 2) can be satisfied by setting the radius of the circle for search to $\gamma(\frac{log(i)}{i})^{\frac{1}{d}}$, $\gamma > 2(1+\frac{1}{d})^{\frac{1}{d}}(\frac{\mu Q_{free}}{\zeta_d})^{\frac{1}{d}}$. d is the dimension of the configuration space and ζ_d is the unit volume parameter of the d dimension. The algorithm proposed inherits the assumption of the search radius. Therefore, the DO-RRT$*$ algorithm is asymptotically optimal. The convergence rate to an optimal solution in the DO-RRT* is discussed below.

The optimal path means that the value of ε is the minimum among ε-collision-free paths. In the optimal path, nodes with the minimum ε are distributed near the obstacle. Therefore, the nodes near the obstacle play an important role in creating the optimal path. In the proposed DO-Extend method, if q_{new} exists in the collision area, it moves to the collision-free area using the gradient descent method. Therefore, the probability that q_{new} existing in the collision area can be expanded is greater than zero ($\mathbb{P}(q_{new} \in Q_{obs}) > 0$). The location of the moved sample is distributed near the obstacle, so that the number of nodes located in the ε-collision-free area is increased. As a result, the ratio of nodes required for the optimal path is increased by the proposed extend method, and the convergence speed is faster than RRT*.

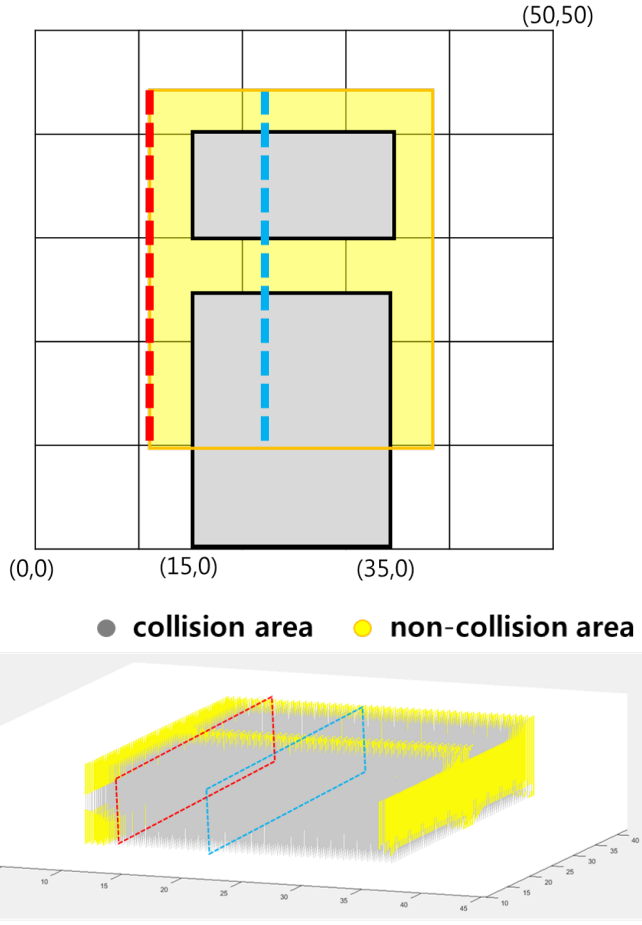


Figure. 4.6: Configuration space analysis

4.5.3 Configuration Space Analysis

In this subsection, the distribution of samples in the configuration space is analyzed proposed sampling method's effectiveness. A workspace with a narrow space can be transformed into a configuration space as shown in Figure.4.6 and the resolution of the configuration space is set to 0.5 m (x,

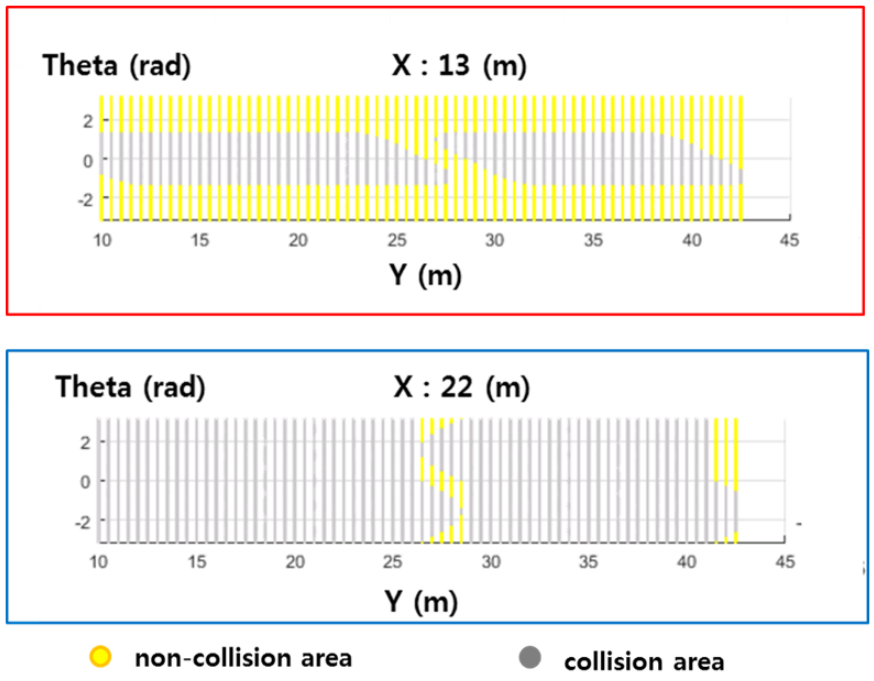


Figure. 4.7: Configuration space analysis for the sagittal plane

y axis) and 1° (θ axis). A yellow area represents the collision-free configurations and grey area represents a configuration collides with obstacles. In Figure.4.7, a $y-\theta$ graph is plotted for a fixed x value (13 m, 22 m).

In the RRT algorithm, nodes that cannot be extended in the random sampling process degrade both the computational performance and path quality. The proposed Desired Orientation-based random sampling method consists of the potential field-based sampling strategy and the magnetic field-based sampling strategy. The potential field-based sampling strategy moves nodes in an expandable direction while taking into account constraints caused by obstacles. We verified the performance of the potential

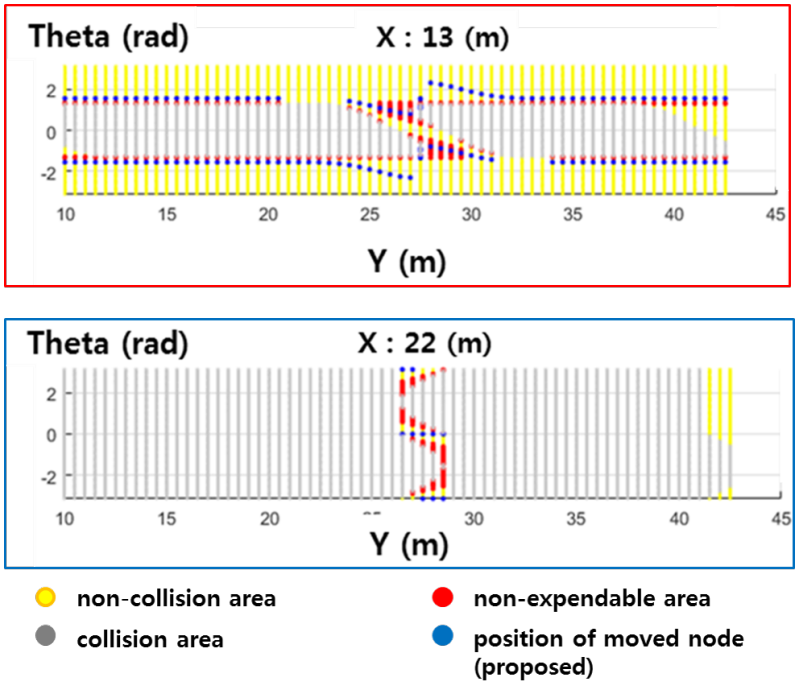


Figure. 4.8: Configuration space analysis of a non-expendable area

field-based sampling strategy by analyzing the unexpanded area in the configuration space and then schematized the moved position of the node according to the application of the proposed method. Figure.4.8 shows the results of calculating the extensibility of nodes in each region. A red area is a region in which obstacles no longer extend the tree. The proposed sampling method plays the role of locating nodes in the blue area. The number of nodes in the non-collision region among the entire region is 14,659. The proposed method reduced the number of nodes in this region by about 10%, and about 13,188 nodes were moved in a feasible direction.

Next, a random sample distribution is analyzed by applying the pro-

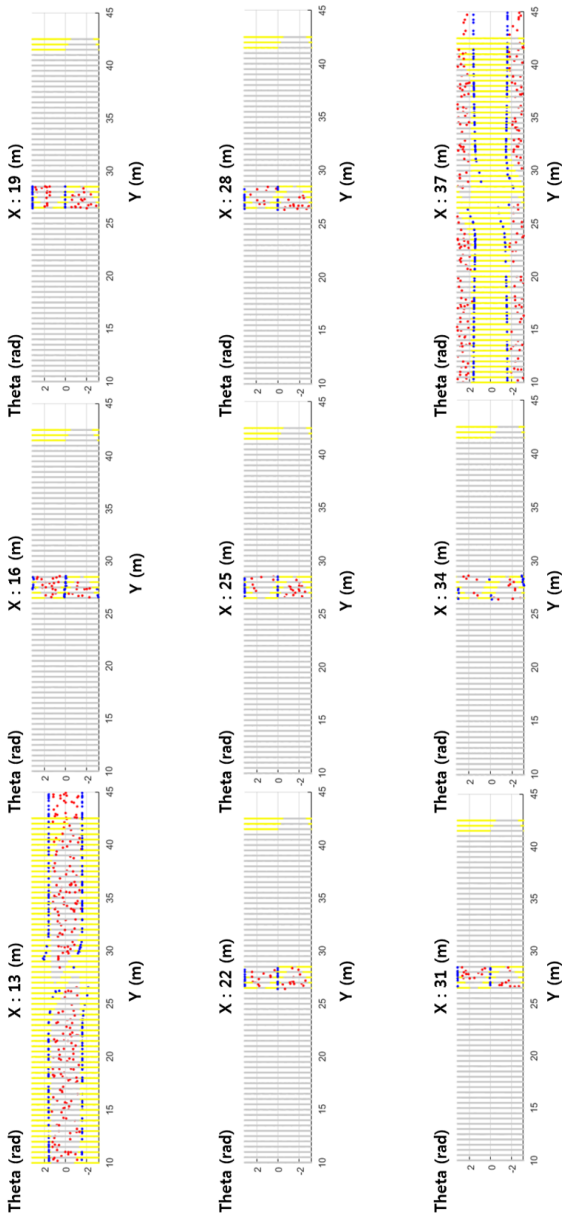


Figure. 4.9: Configuration space analysis of a random sample distribution

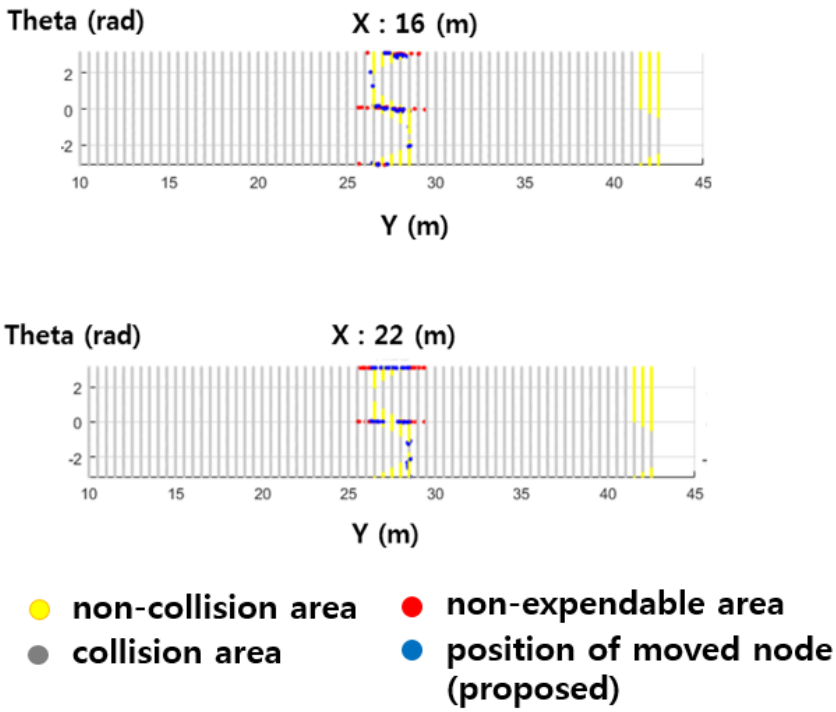


Figure. 4.10:

posed Desired Orientation-based sampling method to the RRT* algorithm. Figure.4.9 is a graph that plots the movement of a node located in a collision area or a non-expandable area when sampling is performed in a workspace. The proposed method has the effect of increasing the number of nodes in a narrow region even though random sampling is performed for invalid regions in the workspace. The proposed Desired Orientation-based extension method moves the q_{new} node to a valid region. In total, 46,655 samples were taken and 7,194 nodes were in the collision area, out of which, 7173 nodes were moved to a valid region within the iteration limit and 21 nodes were

not. Despite the additional computation, the proposed method both enhances the search performance by increasing the number of nodes added to the narrow region and improves the speed of convergence to the optimal path. The numbers of nodes added to the narrow corridor area through RRT$*$ and DO-RRT$*$ are 126 and 1440, respectively. The total number of samplings is 46,655.

4.6 Experimental Results

The DO-RRT and DO-RRT* algorithms are analyzed in a variety of path planning problems in this section. We verified the performance of the DO-RRT and RRT* algorithms by comparing them with the nonholonomic versions of the RRT and RRT* algorithms, which were implemented using a representative library of sampling-based algorithms called the Open Motion Planning Library (OMPL) [51]. The values of experimental parameters and the size of the configuration space were consistent to ensure a proper comparison. In addition, experiments were repeated 50 times for each algorithm with the same samples. The DO-RRT, DO-RRT*, and nonholonomic RRT and RRT* adopted a goal-biased sampling method and a distance metric function based on the Reeds-Shepp algorithm was used [52]. This distance metric function calculates the distance of the Reeds-Shepp path between two SE2 configurations. The Reeds-Shepp path is defined as the shortest path of a nonholonomic car-like robot that can go back and forth with a constrained turning radius. The nearest node can be calculated more effectively than in a Euclidian weight distance method by ensuring the shortest

distance considering both forward and backward movements. The collision detection method is implemented as a k-d tree-based algorithm, as shown in Chapter 3.

4.6.1 Experimental Condition

All Experiments were carried out in a C++ implementation on Intel (R) Core (TM) i5-2500 CPU @ 3.30GHz CPU and 4G Memory and Ubuntu 14.04.3 LTS operating system. Vehicle-specific parameters for the path planning were chosen based on an autonomous vehicle in our laboratory (SPIRIT-1, Hyundai Grandeur). Vehicle length and width are 2845 and 1614 mm, respectively. The distance from the center of the vehicle to the real wheel center is 1100 mm and to the front wheel center is 1745 mm. The maximum wheel angle is 35.0°.

4.6.2 The Result of Desired Orientation-RRT Planner

Experiments were conducted for four scenarios to compare the performance of the path planning algorithms in narrow and cluttered environments: forward parking, backward parking, parallel parking, and free space parking. Each parking scenario was set to require both forward and backward maneuvers due to the narrow space. In contrast to parking scenarios, a simulation in an open area without obstacles was also conducted to confirm the validity of the performance analysis. Figure. shows details of the scenarios.

The performance evaluation parameters are as follows: processing time,

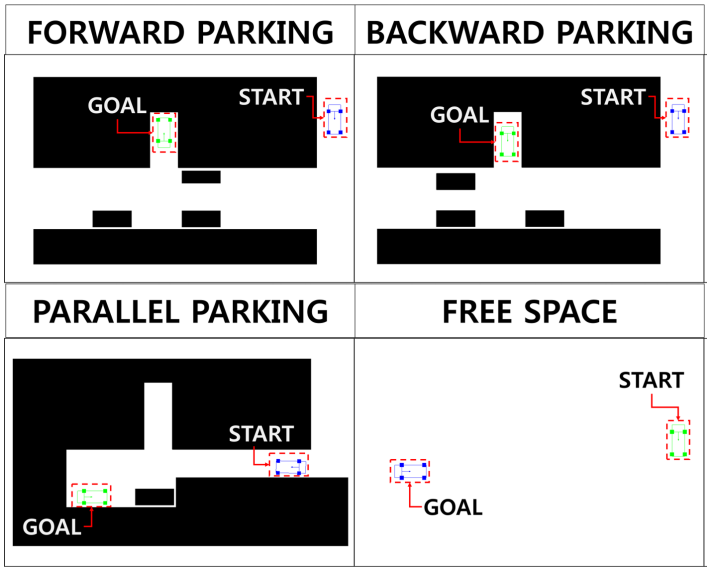


Figure. 4.11: Environments of four scenarios: forward parking, backward parking, parallel parking and free space.

path length, and memory usage. The processing time of the RRT algorithm is the time required to find a feasible path. Since the RRT algorithm does not guarantee path optimality, the path's quality cannot be improved even if the processing time is increased. Therefore, an algorithm that can generate a feasible path in a short time is a key factor. The algorithm was executed for a maximum of 10 s for each scenario. Figure.4.12 shows the simulation results of the DO-RRT and NH-RRT algorithms.

The average processing time of the DO-RRT algorithm was less than the RRT algorithm in the parking scenario and the standard deviation of the processing time spent in the DO-RRT algorithm was also decreased. The processing time was approximately 3.04 times that of the conventional

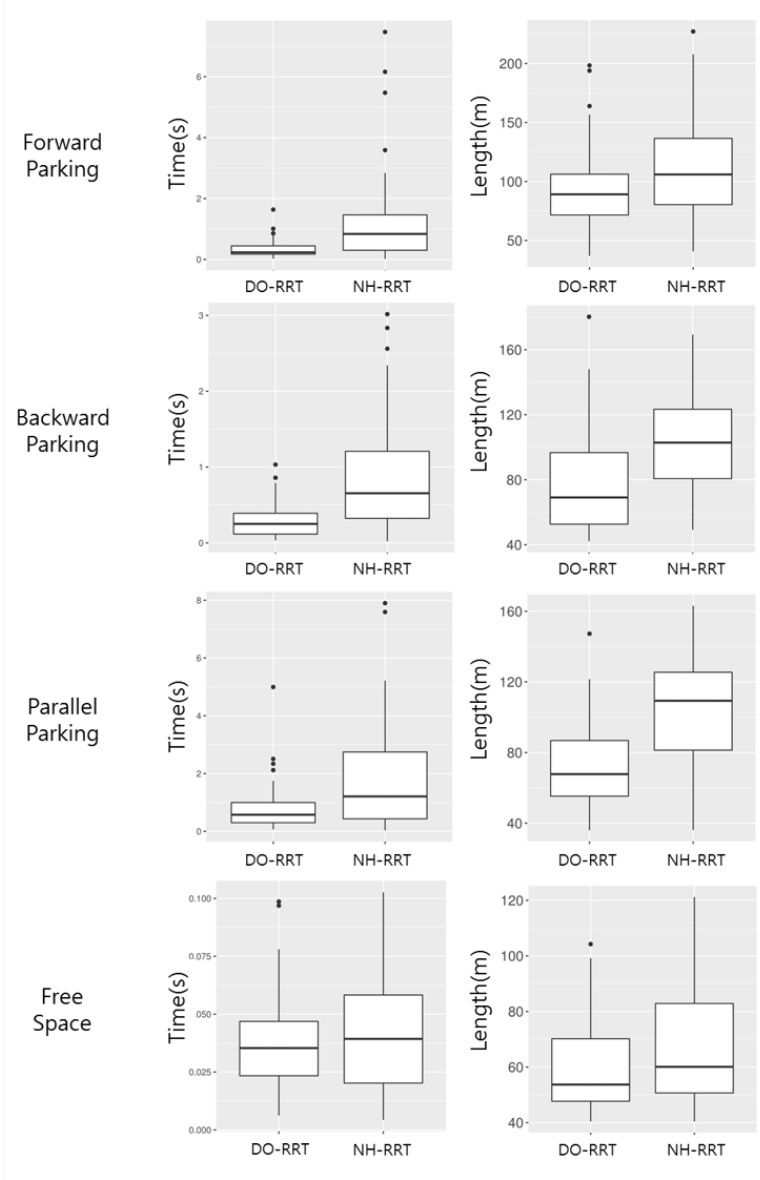


Figure. 4.12: Experimental analysis of DO-RRT and NH-RRT.

RRT algorithm, while the standard deviation was reduced 3.48 times. The path length of the proposed method was decreased by about 1.33 times.

Table 2: Comparision results of DO-RRT and NH-RRT

		Method	Min.	1st	Qu.	Median	Mean	3rd
FORWARD PARKING	TIME (s)	DO-RRT	0.025	0.173	0.233	0.346	0.446	1.638
		NH-RRT	0.017	0.301	0.836	1.257	1.463	7.469
	LENGTH (m)	DO-RRT	36.970	71.580	89.110	93.570	106.200	198.300
		NH-RRT	40.880	80.390	106.000	111.400	136.600	227.100
	MEMORY (MB)	DO-RRT	20.600	21.280	21.280	21.330	21.280	21.750
		NH-RRT	21.750	22.140	23.430	23.740	23.430	27.030
BACKWARD PARKING	TIME (s)	DO-RRT	0.034	0.115	0.250	0.297	0.391	1.031
		NH-RRT	0.023	0.323	0.654	0.871	1.207	3.016
	LENGTH (m)	DO-RRT	42.250	52.680	69.090	76.810	96.670	180.300
		NH-RRT	49.230	80.650	102.800	104.700	123.300	169.300
	MEMORY (MB)	DO-RRT	19.740	20.220	20.220	20.260	20.530	20.530
		NH-RRT	20.530	22.100	22.100	22.030	22.100	22.330
PARALLEL PARKING	TIME (s)	DO-RRT	0.065	0.299	0.574	0.818	0.997	4.996
		NH-RRT	0.040	0.433	1.207	1.805	2.747	7.899
	LENGTH (m)	DO-RRT	36.100	55.310	67.790	72.120	86.820	147.300
		NH-RRT	36.210	81.380	109.300	104.000	125.500	163.100
	MEMORY (MB)	DO-RRT	20.040	20.670	20.790	20.720	20.790	20.790
		NH-RRT	20.790	21.320	21.320	21.510	21.840	21.840
FREE SPACE	TIME (s)	DO-RRT	0.006	0.023	0.035	0.038	0.047	0.099
		NH-RRT	0.004	0.020	0.039	0.040	0.058	0.103
	LENGTH (m)	DO-RRT	40.450	47.690	53.690	59.820	70.190	104.200
		NH-RRT	40.420	50.660	60.080	67.390	82.880	121.200
	MEMORY (MB)	DO-RRT	19.960	20.120	20.120	20.100	20.120	20.120
		NH-RRT	20.120	20.120	20.430	20.370	20.570	20.570

The path from the conventional RRT had many direction changes to align the target position in the narrow region. Thus, the length of the path was increased compared to the DO-RRT algorithm. In the free space scenario, performance degradation of the RRT algorithm caused by obstacles is not a problem. Therefore, performance improvement cannot be expected through the proposed extension method. The execution time is similar to the parking scenario (DO-RRT: 0.037 s, RRT: 0.040 s). However, the proposed sampling method can generate a shorter path length than the RRT algorithm in free space because nonholonomic constraints of the vehicle are considered in the target position. Memory usage is determined by the number of nodes added to the tree during the node expansion process. The proposed method clearly consumes less memory by reducing the number of nodes that cannot

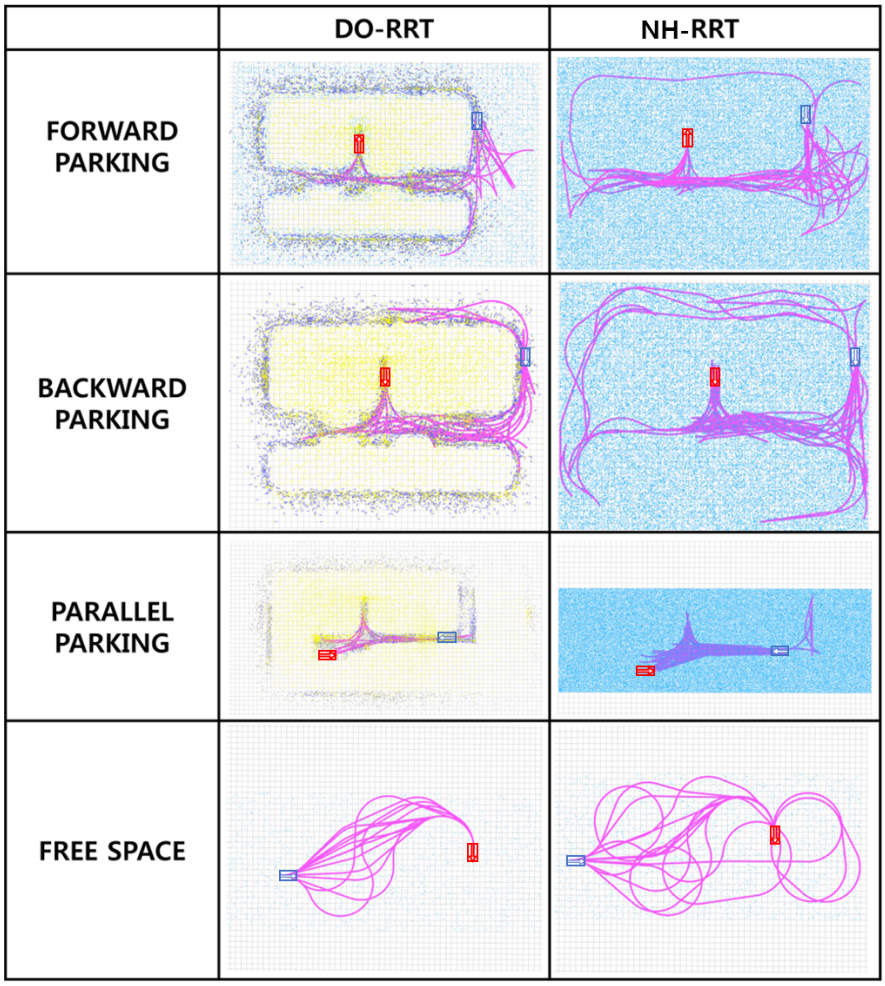


Figure. 4.13: Experimental results of DO-RRT and NH-RRT.

be extended. The proposed method's performance can be improved, as the ratio of the feasible region to the search space is reduced. In this method, additional computation is required in the sampling and extension methods, but it can be inferred that it is more effective by reducing the number of rejected samples. We analyzed the quality of the generated path by measur-

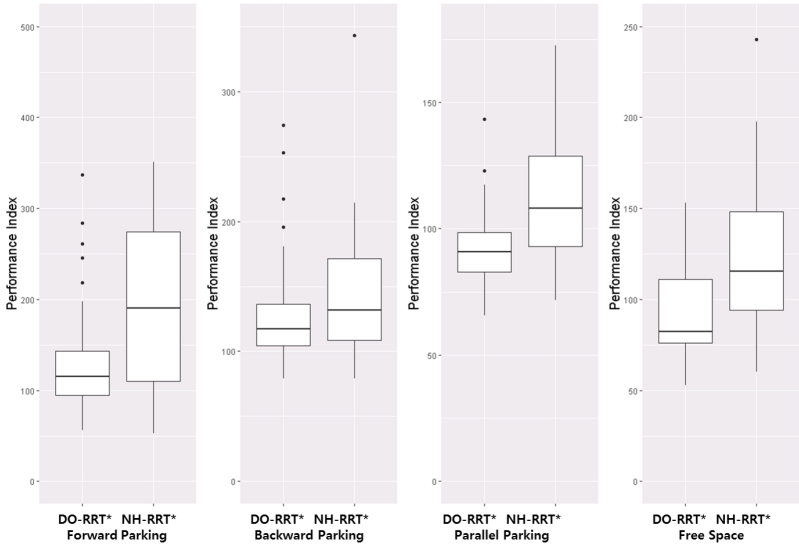


Figure. 4.14: Performance index of DO-RRT* and NH-RRT*

ing the performance index of the generated results. The performance index was calculated by applying weights to the total distance, backward distance, variation of curvature, and number of gear changes. The resultant equation was as follows.

$$P = \alpha L_{forward} + \beta L_{backward} + \delta N_{gearchange} + \gamma \phi_{curvature}. \tag{4.1}$$

Weights were set to $\alpha = 0.719, \beta = 1.438, \delta = 3.0$, and $\gamma = 1.0$. Figure.4.14 is a graph plotting the measurement results of four parking scenarios. The performance index of the proposed method showed a low mean and variance for all scenarios. In other words, it meant that the proposed method could generate a more effective route than NH-RRT.

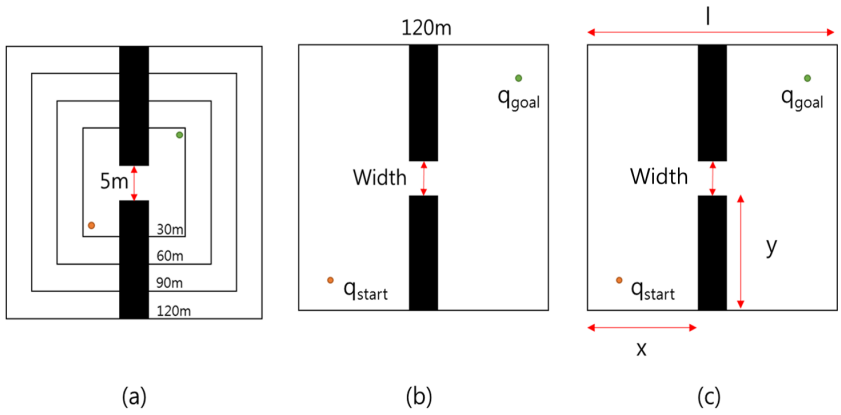
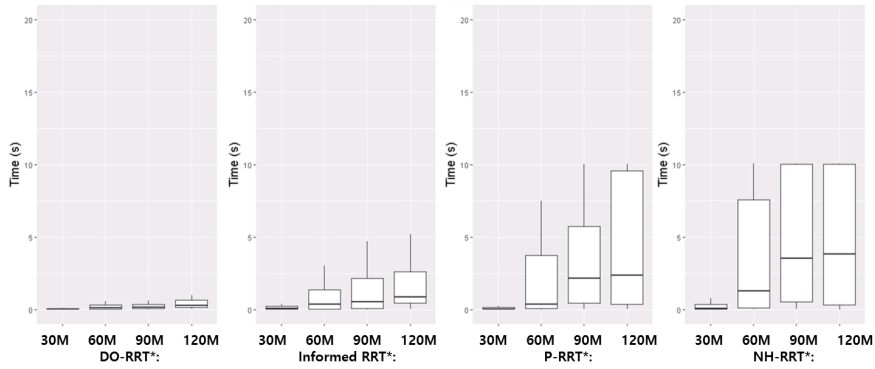


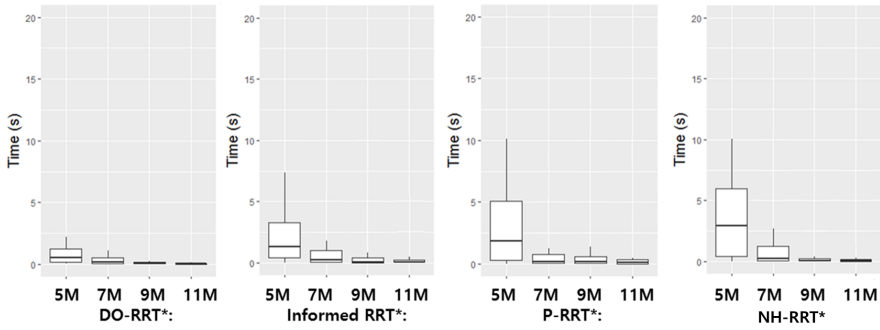
Figure. 4.15: Environments of simple planning problems and randomly generated planning problems

4.6.3 Result of Desired Orientation-RRT*

Unlike the RRT algorithm, the RRT* algorithm calculates the optimal solution asymptotically. An important factor in algorithm performance is the rate at which it converges to the optimal solution in a given time. The performance improvement of the proposed method for a narrow and cluttered environment can be verified by comparing the DO-RRT* method to Informed-RRT*, P-RRT*, and NH-RRT* on a variety of simple planning problems and randomly generated planning problems. In addition, four parking scenarios were analyzed. In simple planning problems, the computation time was measured according to the width of the search space and the gap spacing between obstacles. In a randomly generated problem, the quality of the resultant solution produced during the given time was analyzed by varying the obstacle's location. Figure.4.15 shows environments for analyzing simple planning problems. In Figure.4.15(a), the computation time required to



(a)



(b)

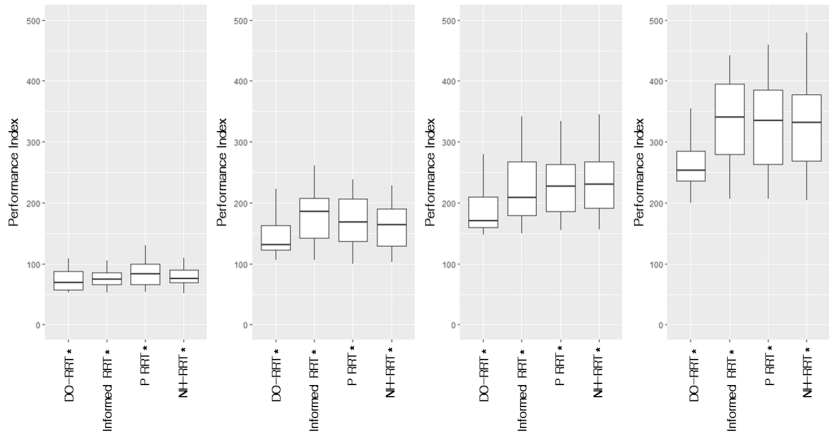
Figure. 4.16: Environments results of simple planning problems.

reach the target cost was measured with respect to search ranges of 30-120 m. Figure.4.15(b) is an environment for measuring the performance from 5 to 11 m in terms of the gap width between obstacles. Figure.4.16 shows the environmental results of simple planning problems. The average time and standard deviation of the proposed method were shortest upon comparison with three other methods in both cases. These average times of 5 m gap spacing (in the search range of 120 m) were DO-RRT*: 0.295 s, Informed-RRT*: 0.850 s, P-RRT*: 2.384 s and RRT*: 3.824 s, respectively.

The standard deviations were DO-RRT*: 0.562s, Informed-RRT*: 3.005s, P-RRT*: 4.170s, and RRT*: 4.579s. Informed-RRT* and P-RRT* could effectively improve the performance depending on the search range, provided there was sufficient free space between obstacles. However, the computation time was significantly increased according to the search range for a narrow gap spacing.

Figure.4.15(c) shows a randomly generated planning problem. This experiment was performed to analyze the quality of the solution path. Unlike previous simulations, obstacles were randomly placed and the analysis of the performance index was performed on a planned optimal path over a given time. In Figure.4.17, the performance index of the result path is compared for a limited time (3 s). Each graph is plotted every 30 m for a search range of 30-120 m. In the proposed method, the performance index average and standard deviation were small for the increase of the search range. In the case of the 30 m search range, the performance difference was not significant because the planning time was sufficient for finding the optimal path. However, as the search range increased, the differences in the performance index clearly showed significant reductions.

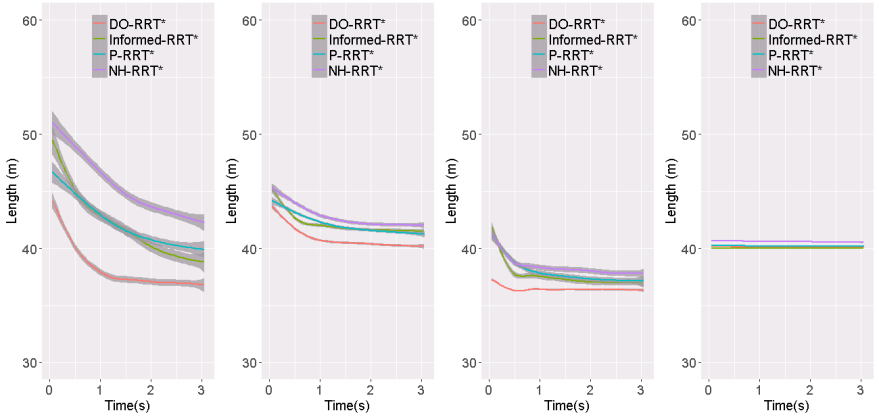
In the parking simulation, as shown in Figure.4.11, DO-RRT*, Informed-RRT* , P-RRT*, and RRT* algorithms were each performed for four scenarios in 3 s. Performance evaluation parameters consisted of the convergence rate and performance index and Figure.4.18 shows the simulation results. The graphs of Figure.4.18(a) are the results of the convergence rate, the line is the mean cost, the shaded area is the mean±SD. The convergence rate to the optimal solution was faster for all parking scenarios and the standard



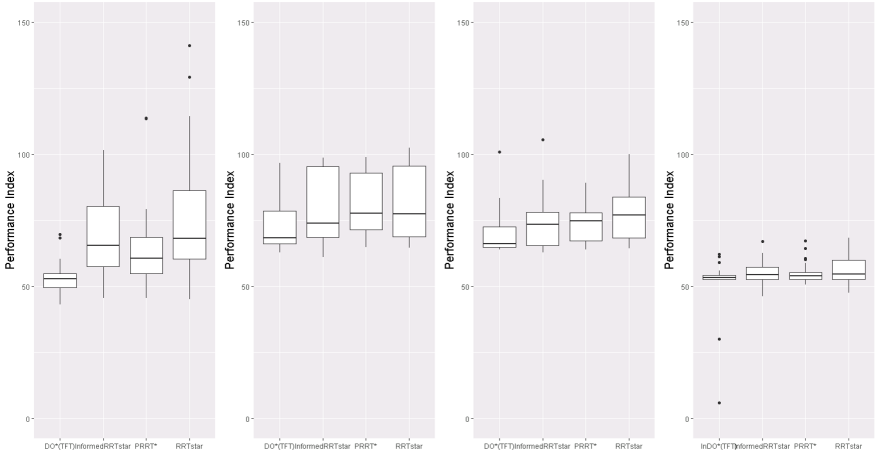
(c)

Figure. 4.17: Environment results of randomly generated planning problem.

deviation of the final cost was also smaller than for other RRT* algorithms (DO-RRT*: 0.7925, Informed-RRT*: 1.682, P-RRT*: 2.068, RRT*: 3.782), indicating that the proposed method is robust. Note that the time and cost required to compute the initial solution were both reduced. Figure.4.18(b) plots the performance index of the result path. The performance index of the proposed method showed a low average value and deviation in all scenarios; this result implies that the proposed method can generate efficient paths over the same amount of time.



(a)



(b)

Figure. 4.18: Environments results of four parking scenarios.

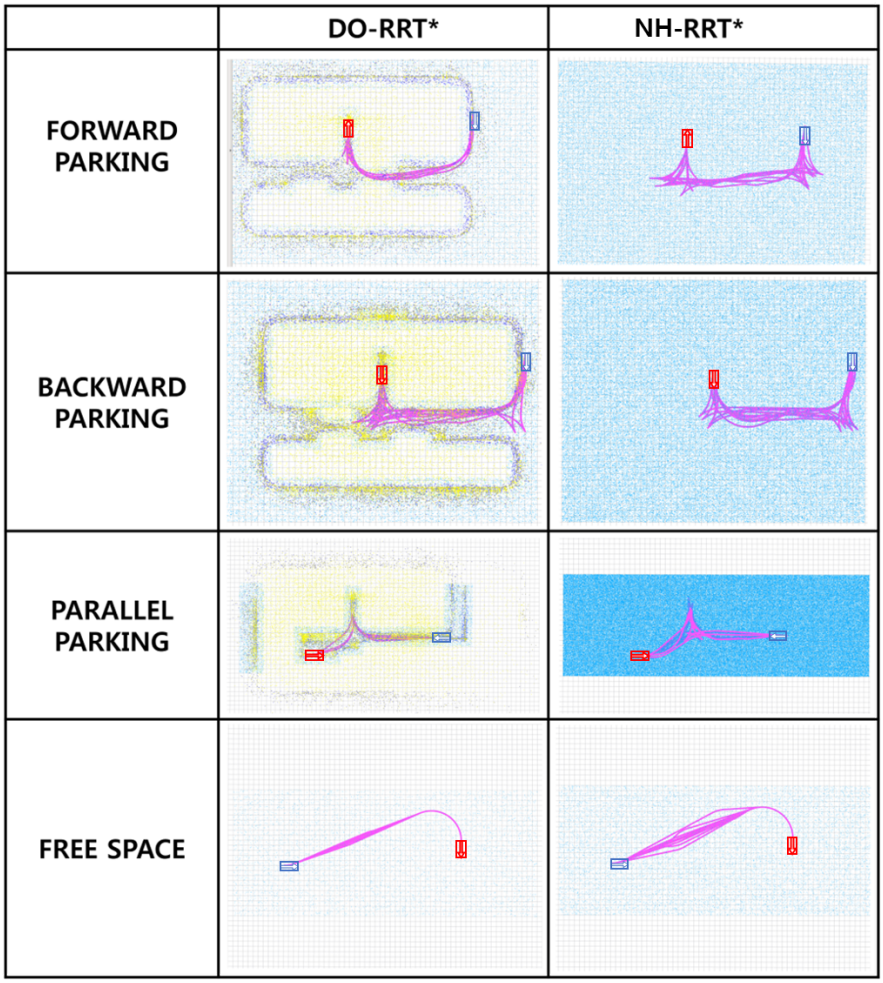


Figure. 4.19: Experimental results of DO-RRT* and NH-RRT*

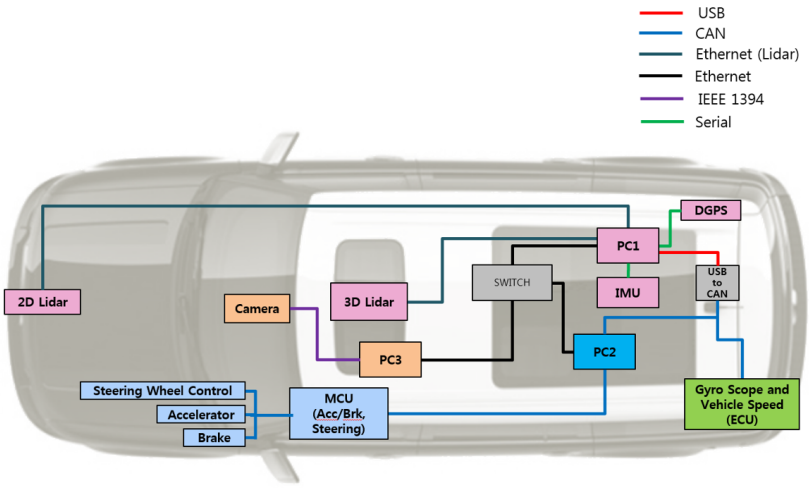
Chapter. 5

Experimental Platform Development

This section deals with the development of an experimental platform for autonomous valet parking; the proposed DO-RRT* method is applied to plan the parking path and is verified through experiments.

5.1 Hardware Architecture

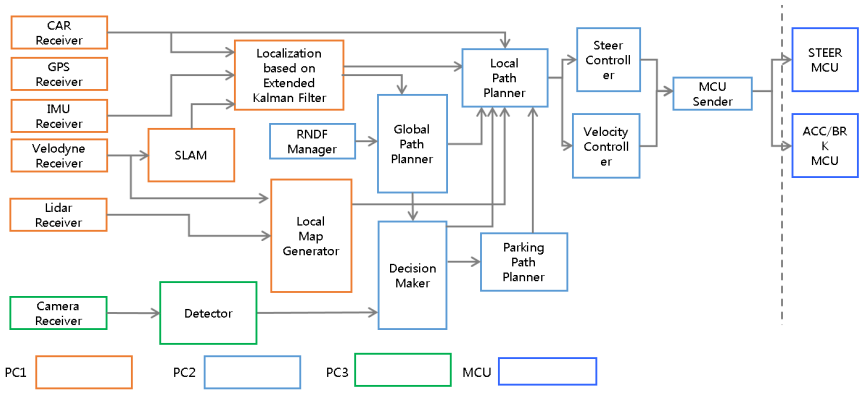
Environment recognition algorithms using Light Detection and Ranging (LIDAR) sensors are performed in PC1. LIDAR sensors consist of Velodyne HDL-64E for 3D data acquisition and SICK LMS 511 for front and rear 2D data acquisition. In addition, a localization algorithm is operated using GPS and IMU sensors to recognize the vehicle's position and orientation. PC2 performs functions related to decision making, path generation, and vehicle controllers and the signal for vehicle control is processed by the additionally mounted MCU. In PC2, the output signal for the control result is transmitted to the MCU via CAN protocol. Three cameras are used for image acquisition and image processing in PC3; the hardware architecture of the autonomous vehicle system is as shown in Figure.5.1.



(a)

(b)

Figure. 5.1: Hardware Architecture (a) and Specifications (b)

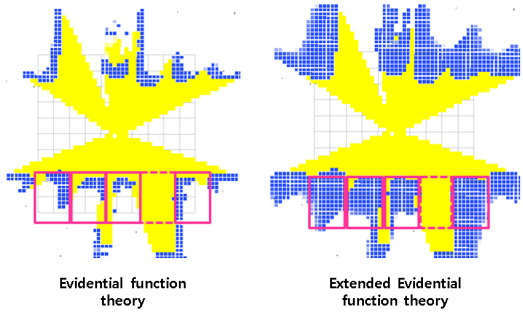


(a)

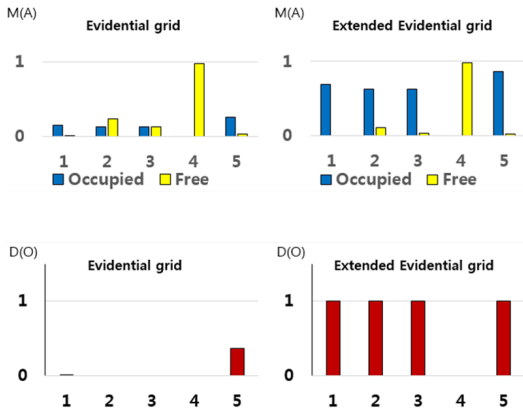
Figure. 5.2: Software Architecture of SPIRIT-1

5.2 Software Architecture

Software for autonomous vehicles runs on Ubuntu-based ROS middleware. Each program performs message delivery through interprocess communication (IPC) as supported by ROS. The software for autonomous vehicles is run on Ubuntu-based ROS middleware and each program performs message delivery through interprocess communication (IPC) supported by the ROS. First, there is a receiver module for each sensor (CAN, GPS, IMU, LIDAR, and camera) and data is collected through this module. All the data is used for perception modules (Detector and local map generator) and localization modules (SLAM and KF-based localization). The planner consists of a global path planner module for mission planning, a decision making module for behavior planning, and a local and a parking path planner for motion planning. The proposed DO-RRT* planner is implemented in the parking path planner module and the controller calculates the target steering angle



(a)



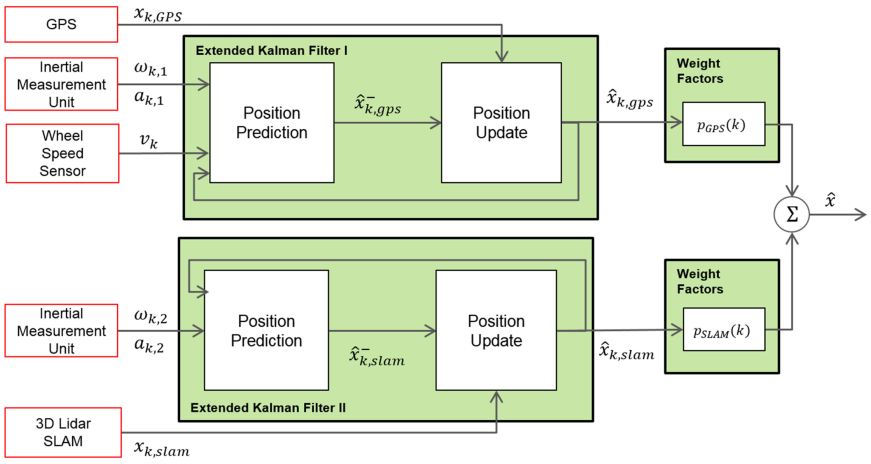
(b)

Figure. 5.3: Extended evidential map based perception system

and speed using the path from the local path planner; this information is sent to the MCU mounted on the vehicle via the MCU send module. Figure.5.2 presents the software architecture of SPIRIT-1.

5.3 Valet Parking System

This section introduces the perception, localization, planning and control systems implemented for an autonomous valet parking system.

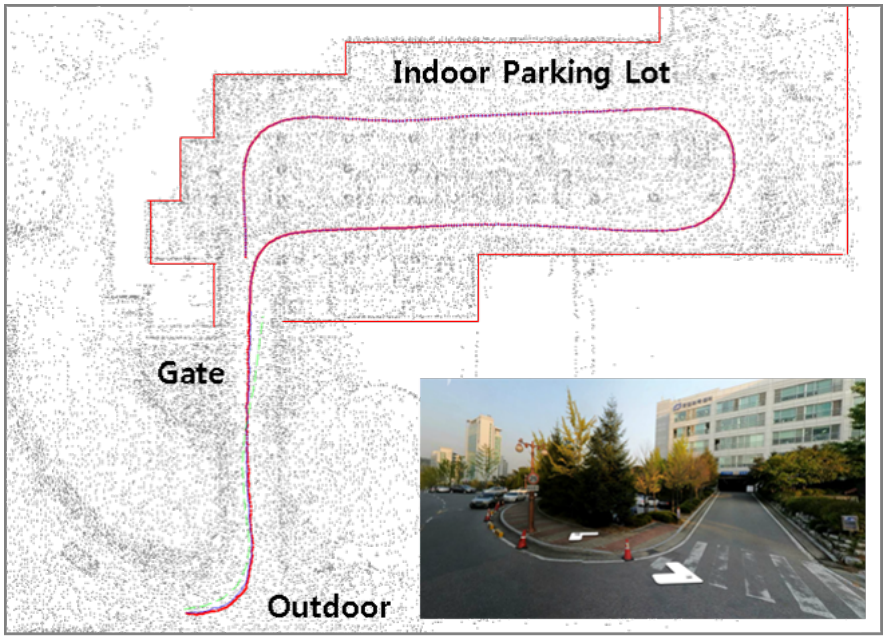


(a)

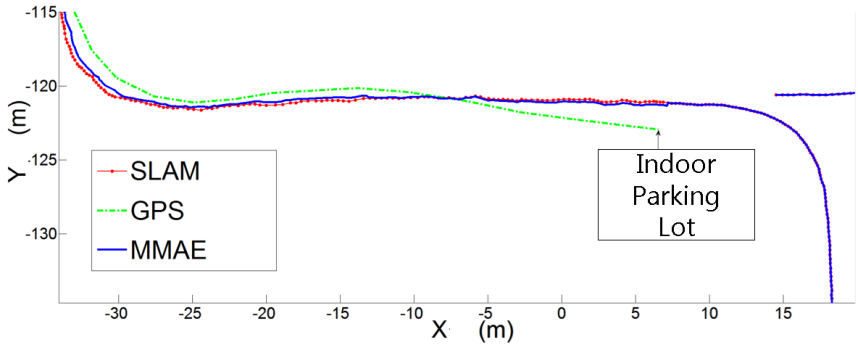
Figure. 5.4: Indoor-Outdoor Localization System based on MMAE method

5.3.1 Perception System

The perception system of the parking area is implemented using the LIDAR-based evidential grid map [53]. The evidence grid map is constructed based on the Dempster–Shafer theory. This map can determine both the occupied and occlusion areas. In the parking area, it is difficult to decide on the parking area due to occlusion. Solving this problem requires that [53] provides four types of probability information for each grid in consideration of the vehicle's kinematic characteristics, and it enables the determination of parking capability. The system generates a parking path using an extended evidence-based map.



(b)



(b)

Figure. 5.5: Results of MMAE method based Localization

5.3.2 Localization System

Vehicle localization technology can be classified as Global Positioning Systems (GPS), Inertial Measurement Unit (IMU)-based methods, and

Simultaneous Localization and Mapping (SLAM) methods to match feature information obtained from cameras or LIDAR systems. GPS-based methods are suitable for measuring vehicles' absolute positions. Therefore, these are widely applied in outdoor driving environments. Although SLAM has the advantage of being applicable to all indoor and outdoor localizations, storing map information in a large space has a high memory requirement and a high computational cost. In addition, positioning errors can be large in locations in which feature extraction is difficult, such as in a free space. The implemented localization system constructs a Kalman filter in parallel to acquire GPS-based location information that is useful for outdoor positioning and SLAM-based location information for indoor positioning. In addition, applying the Multiple Model Adaptive Estimation (MMAE) method [54] allows fusing the measurement values of two Kalman filters to enable indoor and outdoor continuous positioning.

5.3.3 Planning System

Global path planning for autonomous driving can be divided into methods for structured environments and methods for unstructured environments. A global path can be generated using a predefined map for structured environments. In our laboratory, the predefined map was constructed using the Route Network Definition File (RNDF) format, which includes lane, stop line, and parking area information in the Gwanggyo area. The constructed global map is converted into a graph structure for calculating the optimal path. It is implemented so that a global path search can be performed when a user provides a destination location. A directed graph search for an op-

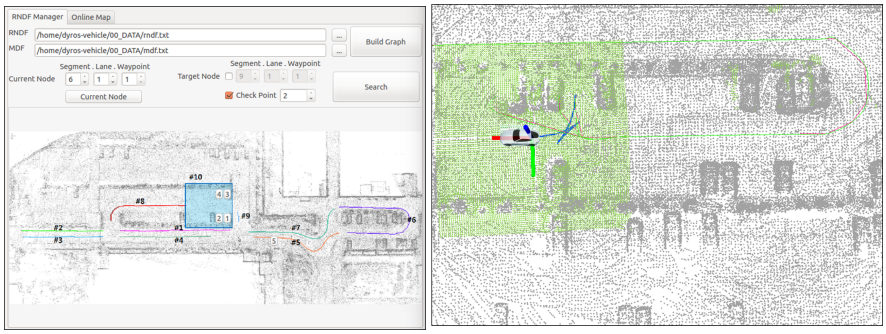
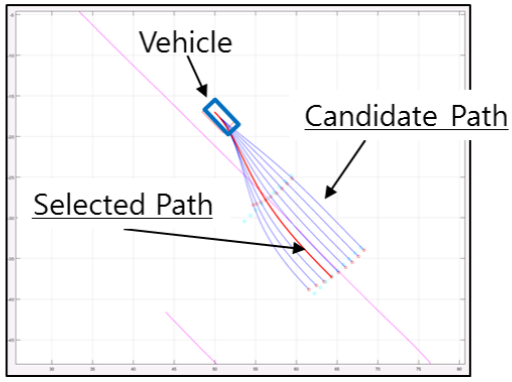


Figure. 5.6: RNDF-based global path planner

timal path is implemented based on Dijkstra's algorithm. Figure.5.6 is the RNDF-based global path planner.

The local path planner computes the vehicle-centric path based on global path information. A spline-based local path planner has been developed to create a safe path that avoids colliding with obstacles. This planner creates candidate paths that satisfy the boundary conditions of the searched global path and the optimal path is selected by calculating the safety, smoothness, and consistency costs of each candidate path.

In an unstructured environment such as a parking lot, it is difficult to create a global route that is based on maps. Therefore, a path should be created that considers the constraints that the surrounding environment place upon the target point. The DO-RRT* algorithm has been applied to solve this problem. The global path should be replanned according to changes in the surrounding environment. Therefore, the path is periodically searched in the background each second. The previous path is updated when a collision with an obstacle occurs, or when a localization or tracking error is



$$J[i] = \omega_s C_s[i] + \omega_\kappa C_\kappa[i] + \omega_c C_c[i]$$

$i: index\ of\ path$
$C_s: cost\ for\ the\ safety\ of\ path$
$C_\kappa: cost\ for\ the\ smoothness\ of\ path$
$C_c: cost\ for\ the\ path\ consistency\ of\ path$

Figure. 5.7: Spline-based local path planner

large. Figure.5.8 shows a flow chart of the planning system for unstructured environments.

5.3.4 Control System

A lateral controller is implemented using a modified version of the pure pursuit algorithm [22]; 5.1 is derived by the geometric bicycle model as illustrated in 5.9. *R* is the radius of the circle that the rear axle will travel along for a given steering angle, *l* is the length of the vehicle, and l_l is the distance between the vehicle's rear axle and the look-ahead point. The look-ahead point is chosen at a predetermined look-ahead distance from the path planning part in consideration of the shortest distance among waypoints.

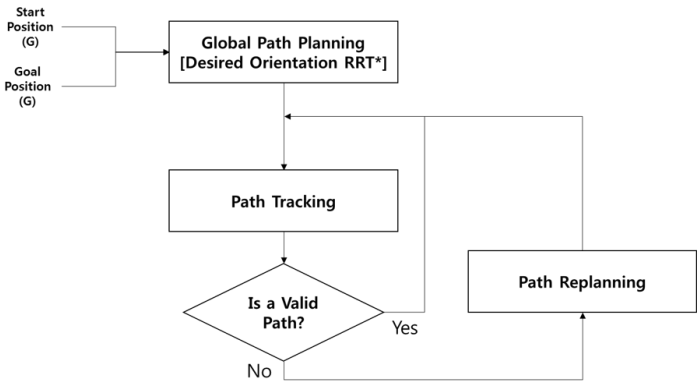


Figure. 5.8: Planning system for an unstructured environement

The steering angle of the vehicle called δ can be determined using the look-ahead point and the angle α between the vehicle's heading vector and the look-ahead vector. The steering angle of the vehicle is determined by the Ackerman steering method with the arc that is calculated by the pure-pursuit algorithm.

$$\delta(t) = \arctan(\frac{2l\sin(\alpha(t))}{l_l}) \tag{5.1}$$

The PID control algorithm is applied to the longitudinal controller and the raw PID control can be expressed as follows

$$\begin{aligned} \alpha(i) &= Kp_a e(i) + Ki_a \int e(i)dt + Kd_a \frac{e(i)-e(i-1)}{\Delta t} \\ \beta(i) &= Kp_b e(i) + Ki_b \int e(i)dt + Kd_b \frac{e(i)-e(i-1)}{\Delta t} \end{aligned} \tag{5.2}$$

where Kp, Ki, and Kd represent the controller gains. The error between the vehicle's current velocity and its desired velocity is denoted by e, α is the term for the acceleration pedal, and the β is the angle of the brake

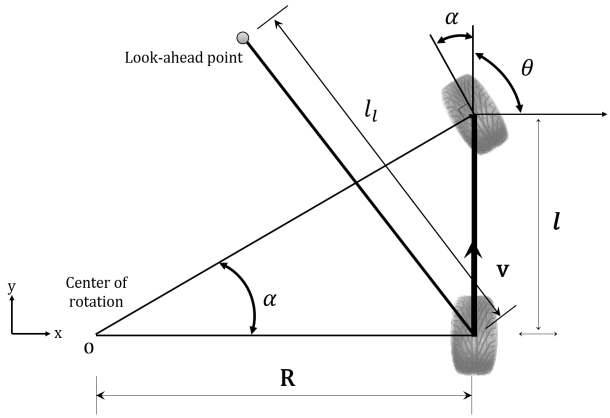


Figure. 5.9: Kinematic bicycle model

pedal's desired position. The tuned values that are obtained via simulation are used for these terms; the desired throttle input and brake that cannot be activated simultaneously are determined using the output of the longitudinal controller.

5.4 Experimental Validation

The valet parking procedure goes as follows. First, the target parking location is selected by the global planning system. The perception system sends the evidential map information to the planning system. In the autonomous parking mode, the parking path is generated using the DO-RRT* algorithm in the stopped state, and the controller then performs steering and speed control to follow the path; the test was conducted in an outdoor parking lot. 5.10 is the comparison result of the DO-RRT* and RRT* algorithms. The proposed method calculated the initial solution faster than RRT* and the

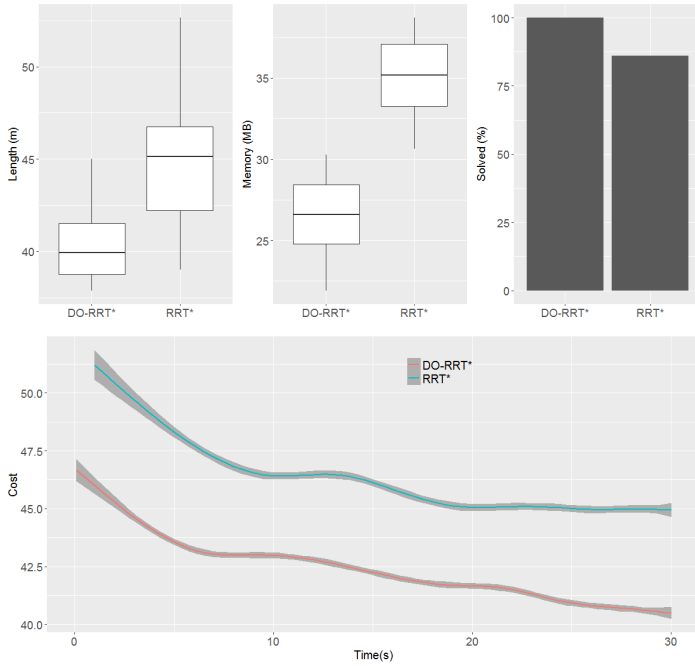


Figure. 5.10: DO-RRT* based valet parking path planning results

path's length was short. DO-RRT* has succeeded in path searches within a limited time frame (30 s), but RRT* has an approximately 10% failure rate. Paths for autonomous parking could be successfully generated using the proposed method.

Figure. 5.11: Results of trajectories generated by DO-RRT* algorithm in a standard parking lot.

Figure. 5.12: Snapshots of the autonomous parking test : (a) Approaching to the Parking lot (b) Shifting gear to the reverse direction (c) Driving backwards toward the parking zone (d) Finishing the parking operation

Chapter. 6

Conclusion

Sampling-based path planning algorithms were discussed in narrow spaces that required many direction changes in both forward and reverse. We modeled the constraints of the vehicle by combining the magnetic-like field and potential fields to calculate feasible directions. In this manner, both environmental constraints and vehicle constraints could be calculated. Thus, unnecessary expansions of the node, which is a typical problem with random sampling methods, can be reduced. In addition, a method for determining the position of the collision point generated in the node's expansion process using the gradient descent method and a method of moving the position of the collision point generated in the node's expansion process to a safe area through the gradient descent method were proposed. This method both increased the efficiency of node expansion and quickly calculated the paths that converge at optimum cost. This was shown by analysis and experimentation applied by both RRT and RRT* methods for the vehicle path generation. The performance of the proposed algorithm has been verified through simulations and actual tests of autonomous vehicles in narrow parking spaces. The RRT method improved the computation speed and reduced memory usage, and the asymptotic optimization method RRT* contributed to improving the optimal solution's calculation speed.

Bibliography

[1] U. S. B. of Transportation Statistics, *Transportation Statistics Annual Report*. Bureau of Transportation Statistics, US Department of Transportation, 2016.

[2] S. Van Themsche, *The Advent of Unmanned Electric Vehicles: The Choices Between E-mobility and Immobility*. Springer, 2015.

[3] S. Kolski, D. Ferguson, M. Bellino, and R. Siegwart, "Autonomous driving in structured and unstructured environments," in *IEEE Intelligent Vehicles Symposium*. IEEE, 2006, pp. 558–563.

[4] U. Ozguner, C. Stiller, and K. Redmill, "Systems for safety and autonomous behavior in cars: The darpa grand challenge experience," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 397–412, 2007.

[5] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[7] A. Stentz, "Optimal and efficient path planning for unknown and dynamic environments," DTIC Document, Tech. Rep., 1993.

[8] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[9] D. Ferguson and A. Stentz, "Field d*: An interpolation-based path planner and replanner," in *Robotics Research*. Springer, 2007, pp. 239–253.

[10] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Resaech*, vol. 5, no. 1, pp. 90–98, 1986.

[11] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.

[12] S. M. Lavalle, "Rapidly-exploring random trees: A new tool for path planning," Computer Science Dept., Iowa State University, Tech. Rep., 1998.

[13] R. Bohlin and L. E. Kavraki, "Path planning using lazy prm," in *IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 2000, pp. 521–528.

[14] L. J. Guibas, C. Holleman, and L. E. Kavraki, "A probabilistic roadmap planner for flexible objects with a workspace medial-axis-based sampling approach," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 1999, pp. 254–259.

[15] J. J. Kuffner and S. M. LaValle, "Rrt-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 2000, pp. 995–1001.

[16] M. Kalisiak and M. van de Panne, "Rrt-blossom: Rrt with a local flood-fill behavior." in *IEEE International Conference on Robotics and Automation*. IEEE, 2006, pp. 1237–1242.

[17] L. Jaillet, J. Cortés, and T. Siméon, "Transition-based rrt for path planning in continuous cost spaces," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 2145–2150.

[18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Resaech*, vol. 30, no. 7, pp. 846–894, 2011.

[19] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, 2001.

[20] D. J. Webb and J. van den Berg, "Kinodynamic rrt*: Asymptotically optimal motion planning for robots with linear dynamics," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5054–5061.

[21] J. L. Blanco, M. Bellone, and A. Gimenez-Fernandez, "Tp-space rrt-kinematic path planning of non-holonomic any-shape vehicles," *International Journal of Advanced Robotic Systems*, vol. 12, 2015.

[22] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105–1118, 2009.

[23] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1961–1976, 2015.

[24] J. hwan Jeon, R. V. Cowlagi, S. C. Peters, S. Karaman, E. Frazzoli, P. Tsiotras, and K. Iagnemma, "Optimal motion planning with the half-car dynamical model for autonomous high-speed driving," in *American Control Conference*. IEEE, 2013, pp. 188–193.

[25] A. Perez, R. Platt, G. Konidaris, L. Kaelbling, and T. Lozano-Perez, "Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics," in *IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2537–2542.

[26] S. Karaman and E. Frazzoli, "Sampling-based optimal motion planning for non-holonomic dynamical systems," in *IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 5041–5047.

[27] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati, and J. H. Reif, "Narrow passage sampling for probabilistic roadmap planning," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1105–1115, 2005.

[28] H.-Y. Yeh, S. Thomas, D. Eppstein, and N. M. Amato, "Uobprm: A uniformly distributed obstacle-based prm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 2655–2662.

[29] I. Ko, B. Kim, and F. C. Park, "Randomized path planning on vector fields," *The International Journal of Robotics Research*, vol. 33, no. 13, pp. 1664–1682, 2014.

[30] L. Palmieri, S. Koenig, and K. O. Arras, "Rrt-based nonholonomic motion planning using any-angle path biasing," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 2775–2781.

[31] B. Akgun and M. Stilman, "Sampling heuristics for optimal motion planning in high dimensions," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 2640–2645.

[32] S. Kiesel, E. Burns, and W. Ruml, "Abstraction-guided sampling for motion planning." in *SoCS*, 2012.

[33] A. H. Qureshi and Y. Ayaz, "Potential functions based sampling heuristic for optimal path planning," *Autonomous Robots*, vol. 40, no. 6, pp. 1079–1093, 2016.

[34] D. Ferguson and A. Stentz, "Anytime rrts," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 5369–5375.

[35] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 2997–3004.

[36] S. Choudhury, J. D. Gammell, T. D. Barfoot, S. S. Srinivasa, and S. Scherer, "Regionally accelerated batch informed trees (rabit*): A framework to integrate local information into optimal path planning," in *IEEE International Conference on Robotics and Automation*. IEEE, 2016, pp. 4207–4214.

[37] D. Hsu, T. Jiang, J. Reif, and Z. Sun, "The bridge test for sampling narrow passages with probabilistic roadmap planners," in *IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 2003, pp. 4420–4426.

[38] J. Lee, O. Kwon, L. Zhang, and S.-E. Yoon, "A selective retraction-based rrt planner for various environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 1002–1011, 2014.

[39] L. Zhang and D. Manocha, "An efficient retraction-based rrt planner," in *IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 3743–3750.

[40] M. Du, J. Chen, P. Zhao, H. Liang, Y. Xin, and T. Mei, "An improved rrt-based motion planner for autonomous vehicle in cluttered environments," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 4674–4679.

[41] M. Rickert, A. Sieverling, and O. Brock, "Balancing exploration and exploitation in sampling-based motion planning," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1305–1317, 2014.

[42] Z. Yan, B. Hao, Y. Liu, and X. Liu, "Dd-rrt path planning and guidance in heading-vector field for a uuv recovery," in *MTS/IEEE OCEANS*. IEEE, 2016, pp. 1–5.

[43] A. Shkolnik, M. Walter, and R. Tedrake, "Reachability-guided sampling for planning under differential constraints," in *IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2859–2865.

[44] S. D. Pendleton, W. Liu, H. Andersen, Y. H. Eng, E. Frazzoli, D. Rus, and M. H. Ang, "Numerical approach to reachability-guided sampling-based motion planning under differential constraints," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1232–1239, 2017.

[45] A. De Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," in *Robot Motion Planning and Control*. Springer, 1998, pp. 171–253.

[46] H. G. Tanner and K. J. Kyriakopoulos, "Nonholonomic motion planning for mobile manipulators," in *IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 2000, pp. 1233–1238.

[47] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[48] D. J. Griffiths and R. College, *Introduction to electrodynamics*. prentice Hall Upper Saddle River, NJ, 1999, vol. 3.

[49] T. Kunz and M. Stilman, "Kinodynamic rrts with fixed time step and best-input extension are not probabilistically complete," in *Algorithmic foundations of robotics XI*. Springer, 2015, pp. 233–244.

[50] S. Karaman and E. Frazzoli, "Optimal kinodynamic motion planning using incremental sampling-based methods," in *IEEE Conference on Decision and Control*. IEEE, 2010, pp. 7681–7687.

[51] I. A. Sucan, M. Moll, and L. E. Kavraki, “The open motion planning library,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.

[52] H. J. Sussmann and G. Tang, “Shortest paths for the reeds-shepp car: a worked out example of the use of geometric techniques in nonlinear optimal control,” *Rutgers Center for Systems and Control Technical Report*, vol. 10, pp. 1–71, 1991.

[53] J. P. Joohee Choi, Seho Shin, “Extended evidential grid map in dynamic environments for an autonomous parking system,” in *Institute of Control, Robotics and Systems*. ICROS, 2016.

[54] J. P. Seho Shin, Joohee Choi, “Development of indoor-outdoor localization system using multiple model adaptive estimation for autonomous vehicle,” in *Korea Robotics Society Annual Conference*. KROS, 2016.

초 록

최근 스마트 자동차 산업이 발전함에 따라 자율 주행 기술과 관련된 연구가 활발하게 진행되고 있다. 차량의 모션 계획은 주행의 안전성 및 효율성과 밀접하게 관계된 핵심적인 기술 중 하나이다. 하지만 협소하고 복잡한 도로에서는 환경적 구속 조건과 차량의 논홀로노믹 구속 조건이 복합적으로 작용되어 모션 계획의 복잡성이 증가된다. 이로 인해 전/후진 전환이 많은 비효율적인 경로를 생성하거나 제한된 시간 내에 경로 탐색에 실패하기도 한다. 본 연구에서는 협소하고 복잡한 환경에서 차량의 경로 생성을 위한 샘플링 기반의 모션 계획 알고리즘을 다룬다. 본 연구의 기여는 다음과 같다. 첫째, 협소한 공간에서의 효율적인 샘플링 방법을 제안한다. 협소하고 복잡한 지역에서는 환경 및 차량의 제약 조건을 만족하는 유효한 샘플의 빈도가 낮고 확장 불가능한 방향의 노드 수가 증가되어 계산 성능을 저하시키고 메모리 사용량을 증가시킨다. 이를 위해 자기장 기반의 접선 벡터 방향 샘플링을 적용한 Desired Orientation Rapidly-exploring Random Trees (DO-RRT)을 제안한다. 둘째, 접선 방향 제약을 기반으로 한 노드 확장 방법을 제안한다. 랜덤 샘플의 방향으로 트리를 확장하는 과정에서 많은 수의 노드가 충돌 테스트를 통과할 수 없다. 이것은 불필요한 샘플링 반복의 횟수를 증가시켜 성능 저하를 야기한다. 이 문제를 해결하기 위해 경사 하강법 기반의 노드 확장 방법을 적용한 DO-RRT* 알고리즘을 제안한다. 제안된 알고리즘의 성능은 다양한 환경

에서 시뮬레이션 및 실차 테스트를 통해 분석되었다.

주요어 : 무인자동차, 모션 플래닝, 경로 계획, 주차 경로, 자율 주행
학번 : 2010-22689